DOCUMENT RESUME

ED 058 896 LI 003 399

AUTHOR	Uemura, Syunsuke
TITLE	Intrex Subject/Title Inverted-File Characteristics.
INSTITUTION	Massachusetts Inst. of Tech., Cambridge. Electronic Systems Lab.
SPONS AGENCY	National Science Foundation, Washington, D.C.
REPORT NO	ESL-TM-454
PUB DATE	Sep 71
NOTE	95p.; (14 References)
EDRS PRICE	MF-$0.65 HC-$3.29
DESCRIPTORS	*Electronic Data Processing; Filing; *Indexes (Locaters); *Indexing; *Information Retrieval; Information Storage; *Information Systems; Subject Index Terms
IDENTIFIERS	*Intrex

ABSTRACT

The characteristics of the Intrex subject/title inverted file are analyzed. Basic statistics of the inverted file are presented including various distributions of the index words and terms from which the file was derived, and statistics on stems, the file growth process, and redundancy measurements. A study of stems both with extremely high and low frequency is also reported. All characteristics are discussed and analyzed in connection with the free vocabulary, in-depth indexing method of Intrex. (Author)

ED 058896

U.S. DEPARTMENT OF HEALTH,
EDUCATION & WELFARE
OFFICE OF EDUCATION
THIS DOCUMENT HAS BEEN REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT. POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL OFFICE OF EDUCATION POSITION OR POLICY

September, 1971 Technical Memorandum ESL-TM-454

INTREX SUBJECT/TITLE INVERTED-FILE CHARACTERISTICS

by

Syunsuke Uemura

The research reported herein was made possible through partial support extended the Massachusetts Institute of Technology, Project Intrex,at the Electronic Systems Laboratory from the National Science Foundation Grant Number GN-774 designated as M.I.T. DSR Project Number 71144.

LI 003 399

Electronic Systems Laboratory
Department of Electrical Engineering
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

ABSTRACT

The characteristics of the Intrex subject/title inverted file are analyzed. Basic statistics of the inverted file are presented including various distributions of the index words and terms from which the file was derived, and statistics on stems, the file growth process, and redundancy measurements. A study of stems both with extremely high and low frequency is also reported. All characteristics are discussed and analyzed in connection with the free vocabulary, in-depth indexing method of Intrex.

ACKNOWLEDGEMENT

The author wishes to express his sincere appreciation to Professor J. Francis Reintjes for giving him the opportunity to study at Project Intrex. He also wishes to extend his hearty thanks to Mr. Richard S. Marcus for guiding him into this field and for providing him with inspiring discussions and suggestions, to Mr. Alan R. Benenfeld for critically reading the draft of this report and for giving many useful suggestions, and to Mr. Robert L. Kusik for the foundation of this work. He is also extremely indebted to Messrs. Charles E. Hurlburt, Michael K. Molnar, and Peter Kugel for their cheerful and patient guidance on the Intrex retrieval system programs.

The author gratefully acknowledges the support given him by the National Aeronautics and Space Administration, and by the Science and Technology Agency, Japan, whose fellowships made it possible for him to study at Project Intrex.

CONTENTS

1.	INTRODUCTION	page 1
2.	GENERAL CHARACTERISTICS OF THE INVERTED FILE	4
	2.1 Introduction	4
	2.2 File Growth Process	4
	2.2.1 Comparison of Four Stages of File Growth	4
	2.2.2 The Number of Stem Types	6
	2.2.3 Stem Occurrences	6
	2.3 Stem Frequency Distribution	9
	2.4 Stem Length Distribution	11
	2.5 Stem Attributes	14
	2.5.1 Introduction	14
	2.5.2 Range Number	15
	2.5.3 Word Number	17
	2.5.4 Term Number	21
	2.6 Correspondance between Stem and Word	23
	2.7 In-depth Indexing and Redundancy	25
	2.7.1 Introduction	25
	2.7.2 Redundancy Measurement	27
	2.7.3 Reduction in Redundancy	28
3.	STEMS WITH HIGH FREQUENCY	35
	3.1 Introduction	35
	3.2 High Frequency Stems	35
	3.3 Change in High Frequency Stem Order	42
	3.4 Redundancy of High Frequency Stems	47

CONTENTS (Cont'd)

		page
	3.4.1 The Number of Distinct Document References	47
	3.4.2 Redundant Usage and Word Meanings	50
	3.4.3 Redundancy Reduction in High Frequency Stems	54
	3.5 High Frequency Common Stems	54
4.	STEMS WITH EXTREMELY LOW FREQUENCY	57
	4.1 Introduction	57
	4.2 Generation of Extremely Low Frequency Stems	57
	4.2.1 File Updating and New Stems	57
	4.2.2 Classification of New Stems	58
	4.3 Extremely Low Frequency Stems in the Whole Inverted File	67
5.	SUMMARY	71
	REFERENCES	75
	APPENDIX A LSTDST Program Listing	77
	APPENDIX B IFSTAT Program Listing	80
	APPENDIX C IFDOCS Program Listing	83
	APPENDIX D MIFLST Program Listing	85

LIST OF FIGURES

2.1	Number of Stem Types as a Function of the Document Collection Size	page 7
2.2	Number of New Stem Types per Document as a Function of Document Collection Size	8
2.3	Stem Frequency Distribution	10
2.4	Length Distribution of Stem Types	12
2.5	Cumulative Percentage of Stem Types as a Function of Stem Length	13
2.6	Range Number Distribution	16
2.7	Word Number Distribution	18
2.8	Index Term Length Distribution	20
2.9	Term Number Distribution	22
2.10	Stems that Represent more than 15 Word Types	26
2.11	Number of Subject Index Words or Stem Types for Each Document Interval	29
2.12	Redundancy Ratio (RR_i) for Each Document Interval	30
3.1	Reduncancy Ratio RR_s for High Frequency Stems	55

1. INTRODUCTION

The purpose of this report is to make clear various characteristics of the Intrex subject/title inverted file and to provide useful information for future system design. Statistics gathered from the inverted file are presented and analyzed along with a study of stems with high frequency and with extremely low frequency. Statistics on stems, the file growth process, stemming procedure, and redundancy measurement are included.

The reader should note the following points:

1. The Intrex retrieval system uses two inverted files, one for subject index terms and titles combined, the other for author names. Only the former inverted file was investigated and the expression "inverted file" in this report always implies the "subject/title" inverted file. Because this is a combined file, most statistics here relate to the index set composed of both index terms plus titles. Data referring to index terms alone, or to titles alone, is specifically stated as such.

2. The Intrex inverted file does not contain 13 common words, which are called out of subject terms and titles during the inverted file generation process. These common words amount to 20 percent of the original subject term plus title words or 25 percent of the remaining (non-common) words. The data contained in this report excludes these common words except as specified. All data are from the inverted file itself.

3. The Intrex retrieval system is based upon "stems", not words. All words, both subject index and title words are stemmed. A stemming algorithm automatically generates for each word both a stem and an ending. Consequently the number of word tokens (or word

occurrences) is equal to the number of stem tokens (or stem occurrences). (Sometimes a word may not be stemmed at all in which case it can be considered as a stem and a null ending.) On the other hand, since several different words can have the same stem, the number of stem types is less than the number of word types. In the following discussion, when "the number of stems" or "the number of words" is referred to, it implies the number of stem tokens or word tokens, respectively. When the subject is "type", it is always so specified.

4. The author tried to avoid using such general terminologies as "lists" or "references", because they are sometimes misleading, especially for those who are not familiar with their definitions in the Intrex file system. Here, "the number of lists" is expressed as "the number of stem types", and "the total number of references" as "the total number of stem occurrences".

5. The data in this report are those for an inverted file associated with a collection of 15, 845 documents. The size of the Intrex data base is continuously growing.

6. A few words are in order about the Intrex indexing procedures and the resultant index terms from which the inverted file is derived. The Intrex indexing methodology uses a free-vocabulary (generally these are words used by the author) to create a set of index terms. Each term generally consists of one or more phrases connected grammatically to form a nominal construction and containing sufficient contextual information for the term to stand-alone and to be a readable, in-depth representation of some aspect of the document.

> Therefore, Intrex index terms are definitely not simple subject headings nor are they lists of keywords or descriptors. Each term is assigned a range number which indicates the extent to which the document devotes itself to the topic described by the term. The total set of index terms and their range numbers gives an in-depth description of the total document. A document currently entering the Intrex system is assigned about eight index terms with each term containing about a dozen words.

As mentioned in item 3 above, the Intrex inverted file is based upon the stems of individual significant words contained in all of the index terms plus the title. (Intrex treats titles essentially as another index term). However, various characteristics of the index term containing a particular occurrence of a stem are stored with the reference to that stem occurrence in the inverted file. These attributes include the range number assigned to the full term, the word position in that term, a term number, the word ending that the stem occurrence had, plus some additional attributes of the document itself such as its being a professional article on original research. The index word stems, and their additional stored attributes differentiate the Intrex inverted file from a simple keyword file, just as the Intrex indexing methodology in constructing terms differentiates it from a simple selection of free descriptor keywords. Unlike some other kinds of inverted files which contain the full index expression (e.g., subject headings, chained terms), an Intrex user wishing to see the full readable index term(s) in which these stems occur must obtain them directly from the stored catalog record rather than from the inverted file index to the record. In the interactive Intrex retrieval system, this is an easily accomplished procedure.

The reader is referred elsewhere for more information about the general Intrex system,[1] the indexing methodology,[2] the inverted file organization,[3] the stemming algorithm,[4, 5] and the interactive interface between the user and the system.[6]

2. GENERAL CHARACTERISTICS OF THE INVERTED FILE

2.1 Introduction

Computer programs were developed and run to gather statistical information from the inverted file (see Appendix for program listings.) Basic figures for the current inverted file are;*

Number of documents	15,845
Number of stem types	27,291
Number of word types	35,645
Total number of stem occurrences	1,204,282
Average number of subject index plus title words per document	76.0
Average number of stem types per document	39.1

The following sections will discuss specific features of the inverted file statistics.

2.2 File Growth Process

2.2.1 Comparison of Four Stages of File Growth

Numeric comparison between four stages of the inverted file is shown in Table 2.1. Data for the 955 document collection (1968) was gathered not only from the file listing itself but also from other sources.[7,8]

---

* As stated in Chapter 1, these values do not include usages of 13 common words.

Table 2.1

Numeric Comparison between Four Stages of
Inverted File Growth

Characteristic	Inverted File Stage			
	May, 1971	Nov. 1970	Sept. 1969	1968
Number of Documents	15, 845	13, 517	8, 235	955
Number of Stem Types	27, 291	25, 314	19, 291	6, 700*
Total Number of Stem Occurrences	1, 204, 282	1, 076, 163	769, 220	99, 600*
Average Number of Word Occurrences per Document	76.0	79.6	93.4	104*
Average Frequency of a Stem Type	44.1	42.5	39.9	15*
Frequency of the most Frequent Stem**	16, 824	15, 698	12, 077	1, 533
Number of Stem Types that have Frequency of 5 or less	20, 513 (75%)	18, 915 (75%)	13, 967 (72%)	4, 500* (67%)
Number of Stem Types that have Frequency Greater than 1, 000	260 (0.95%)	226 (0.89%)	144 (0.75%)	3 (0.04%)

* Rounded values.

** "magnet-" is the most frequent stem in each case. See Section 3.2.

2.2.2 The Number of Stem Types

Because of the "limited" vocabulary in technical papers, it is expected that the number of stem types would not increase as much as the number of documents. This is true, as can be seen in Table 2.1. While the number of documents grew by a factor of 17 (from 955 to 15845), the number of stem types increased only by a factor of 4. Figure 2.1 shows the growth of the number of stem types as a function of the size of the document collection. The slope is becoming gentler but it is not yet flat. Figure 2.2 shows the rate of new stems per document as a function of document collection size. Even now that the collection size exceeds 15,000 documents, an average of 0.85 new stems are being added to the inverted file per new document. Currently, for every 100 new documents added to the data base, about 85 new stems will have to be added to the inverted file.

This is due to the fact that Intrex uses a free indexing vocabulary rather than an authority list of index terms. The indexers can use any proper words for indexing (and users can too for retrieval). In addition:

(a) A certain number of new stems can and do appear in the forms of, for example, proper nouns or formulae.

(b) The system does not incorporate an automatic check and validation of spellings. Misspelled words will become "new" stems. (It should be pointed out that an authority list of terms can be used as an "authorized spelling checker.")

An analysis of new stems is given in detail in Chapter 4, where proper nouns and formulae are shown to be their main source.

2.2.3 Stem Occurrences

The total number of stem occurrences is naturally growing too, but the growth rate is not as high as that for the number of documents. If the indexing procedure is stable and the number of index words per document does not deviate greatly, both growth rates should be almost

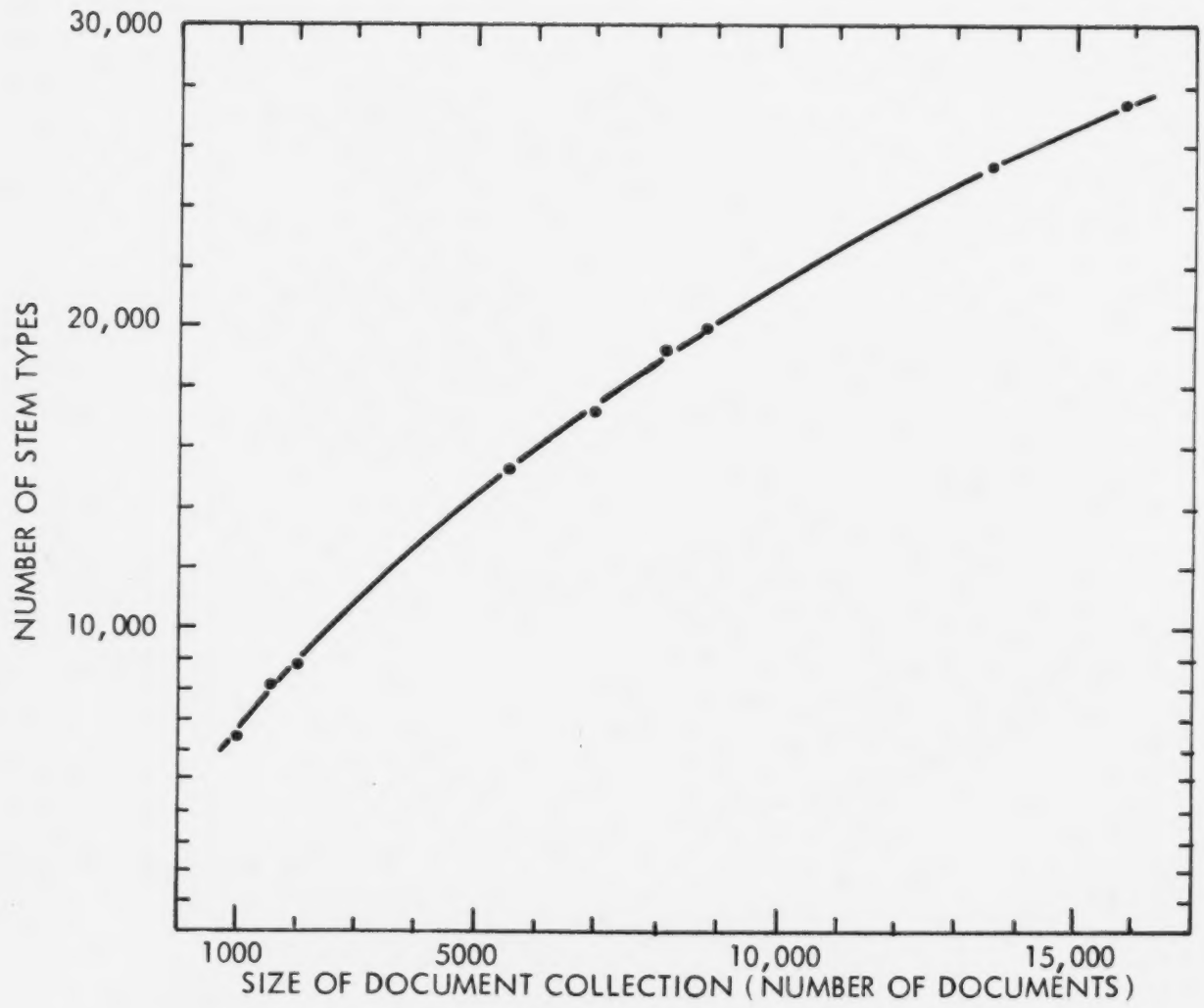

Fig. 2.1 Number of Stem Types as a Function of the Document Collection Size

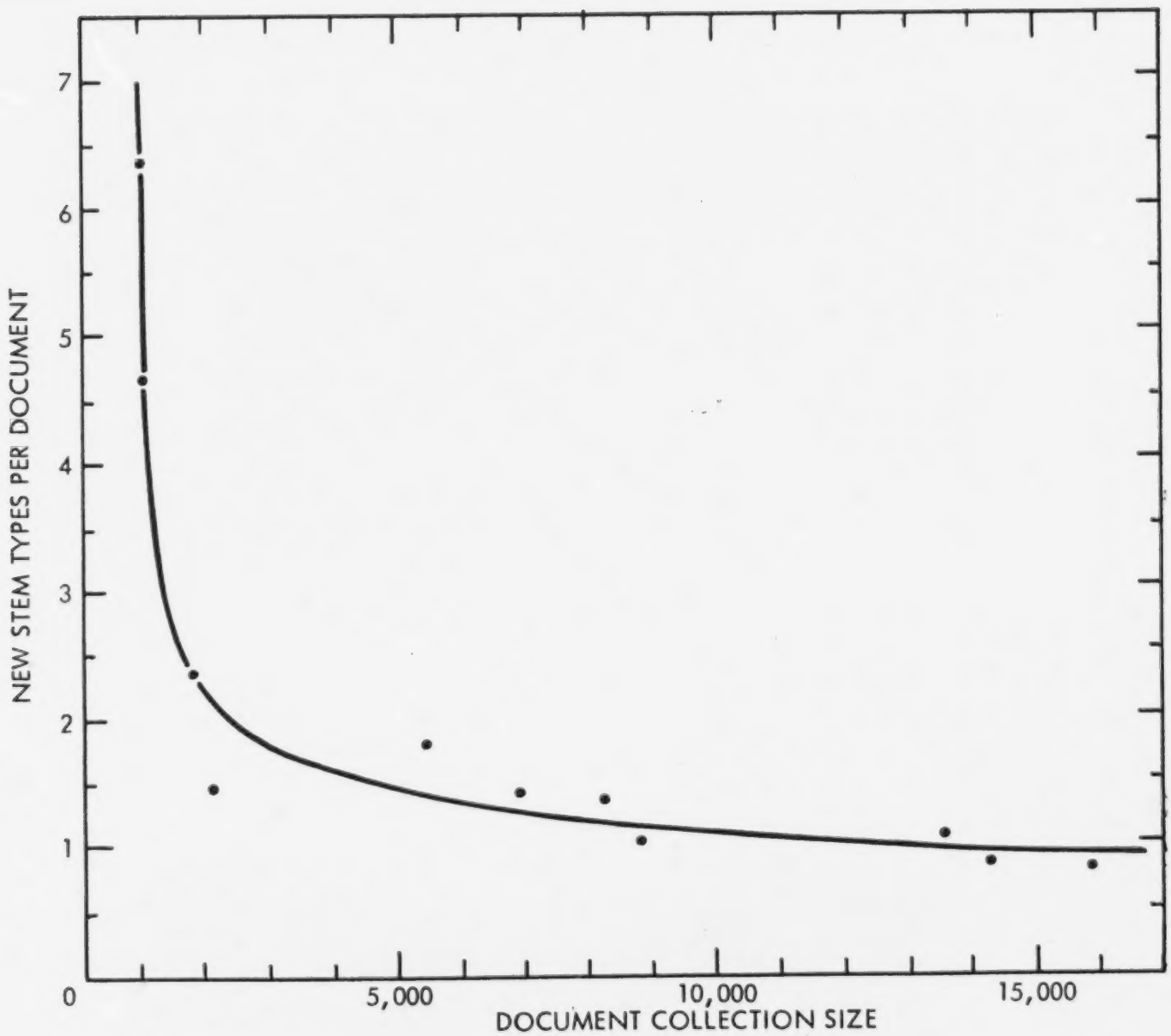

Fig. 2.2 Number of New Stem Types per Document as a Function of Document Collection Size

identical. As a matter of fact, the average number of index words per document is decreasing. It is now 76, formerly more than 100. This is mainly a result of a major change in indexing procedure in 1968, when the indexing method was improved to reduce word redundancy. The inverted file is still redundant. One stem may possibly be used several times in an index. Further study of this redundancy problem appears in Section 2.7.

The average number of stem occurrences per stem type (i.e., stem frequency) is 44.1. Naturally the number is steadily increasing. The value means one stem in the inverted file is assigned an average of 44.1 document numbers. Since some of them are duplicated, the actual number of documents referred to by a stem is less than 44.1. Dividing this value by the redundancy ratio of 1.95 for the whole inverted file (see Section 2.7.3 for a definition and calculation), then the average number of documents referred to by a stem is 22.6. This also means that a user who currently issues a search command with one subject word, will retrieve, on the average, 22.6 documents.

2.3 Stem Frequency Distribution

Figure 2.3 show the log-log plot of the stem frequency distribution for the 15,845 document collection. The distribution for different stages of data base growth can be found elsewhere.[7,9] Except for displacement from an increase in values, the general shape of the frequency distribution remains almost unchanged. There are two extreme types of stems, those with low frequency and those with high frequency.

(1) Stems used only once:
13,085 out of a total 27,291 stems have been used only once for indexing. When the data base contained only 955 documents, it was natural to have many stems each of which had been used only once (1611 stems out of a total 6444 stems). But the number of stems of this type, instead of decreasing by being used more than once, continued to grow as the data base grew. These stems are not particularly useful to users because a search command for one of those

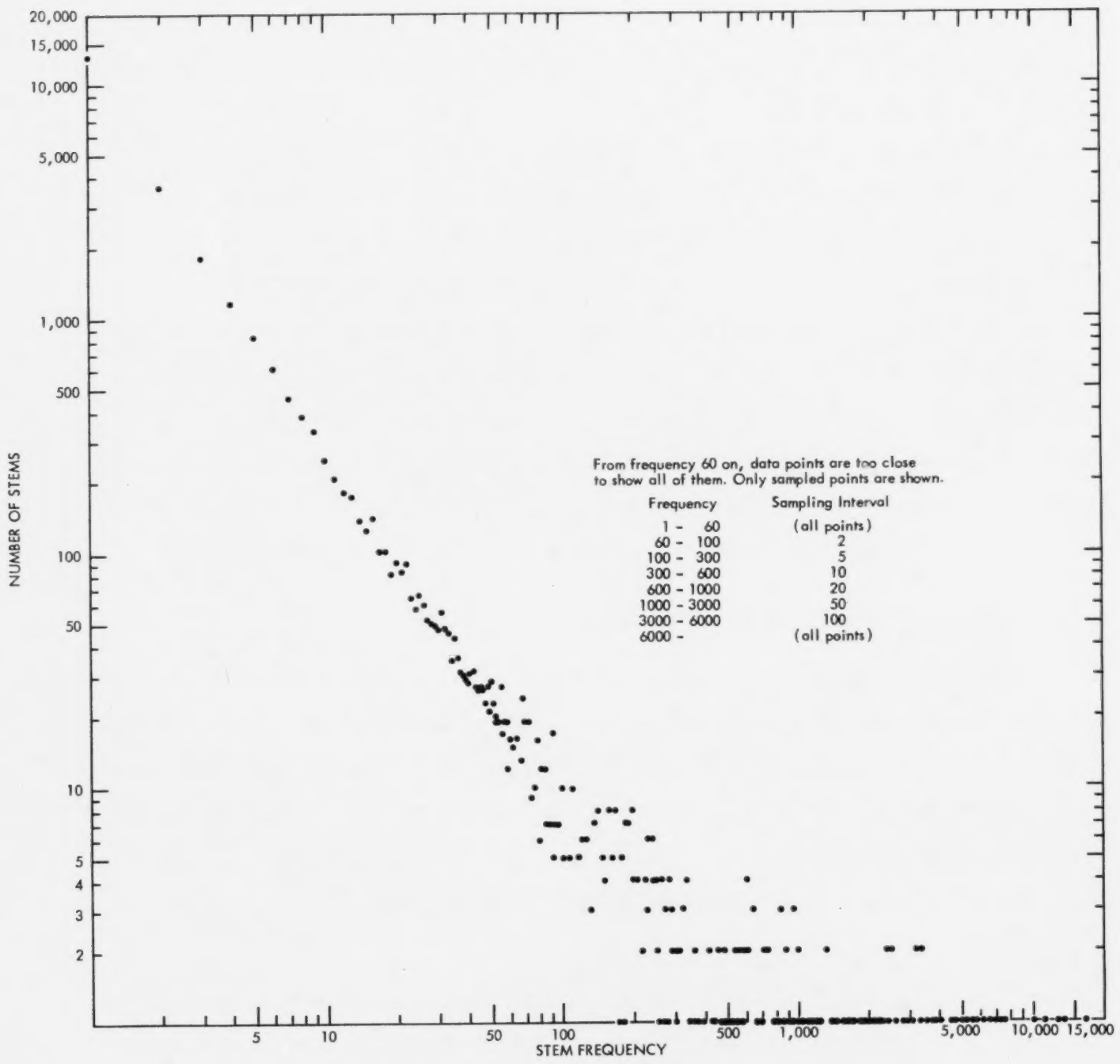

Fig. 2.3 Stem Frequency Distribution

stems, will be able to retrieve only one document. Some of these stems would, of course, be misspellings, some numbers and some formulae. As mentioned in Section 2.2.2, entirely new stems continue to appear in the incremental data base, and many of those new stems may be of this type. Further analysis and discussion of this type of extreme stem appears in Chapter 4.

(2) The high frequency stems:
The stem with the highest frequency is "magnet-", used 16,824 times. There are 260 stems each of which has been used more than 1000 times and they occupy about one half of the entire inverted file space. The user who issues a search command with one of those stems will be forced to restrict his scope by other search terms, since the original stem will retrieve too many documents for him. Further analysis and discussion of this type of extreme stem is given in Chapter 3.

2.4 Stem Length Distribution

Figures 2.4 and 2.5 show stem length distribution. The data here are based upon the length of each stem type; the frequency of each stem type was not taken into account. The average stem type length is 7.76 characters. This is more than two characters longer than the average length of general English words. The reasons are:

(1) The average length of an English word is based upon word frequency distributions and not word types as is the case here. This expected difference between Intrex stem type length and English word frequency length are brought into better focus when we realize that some 20% of all words appearing in the index terms before inverted file generation consist of only 13 common word types only 1 to 3 characters in length (see Table 3.7).

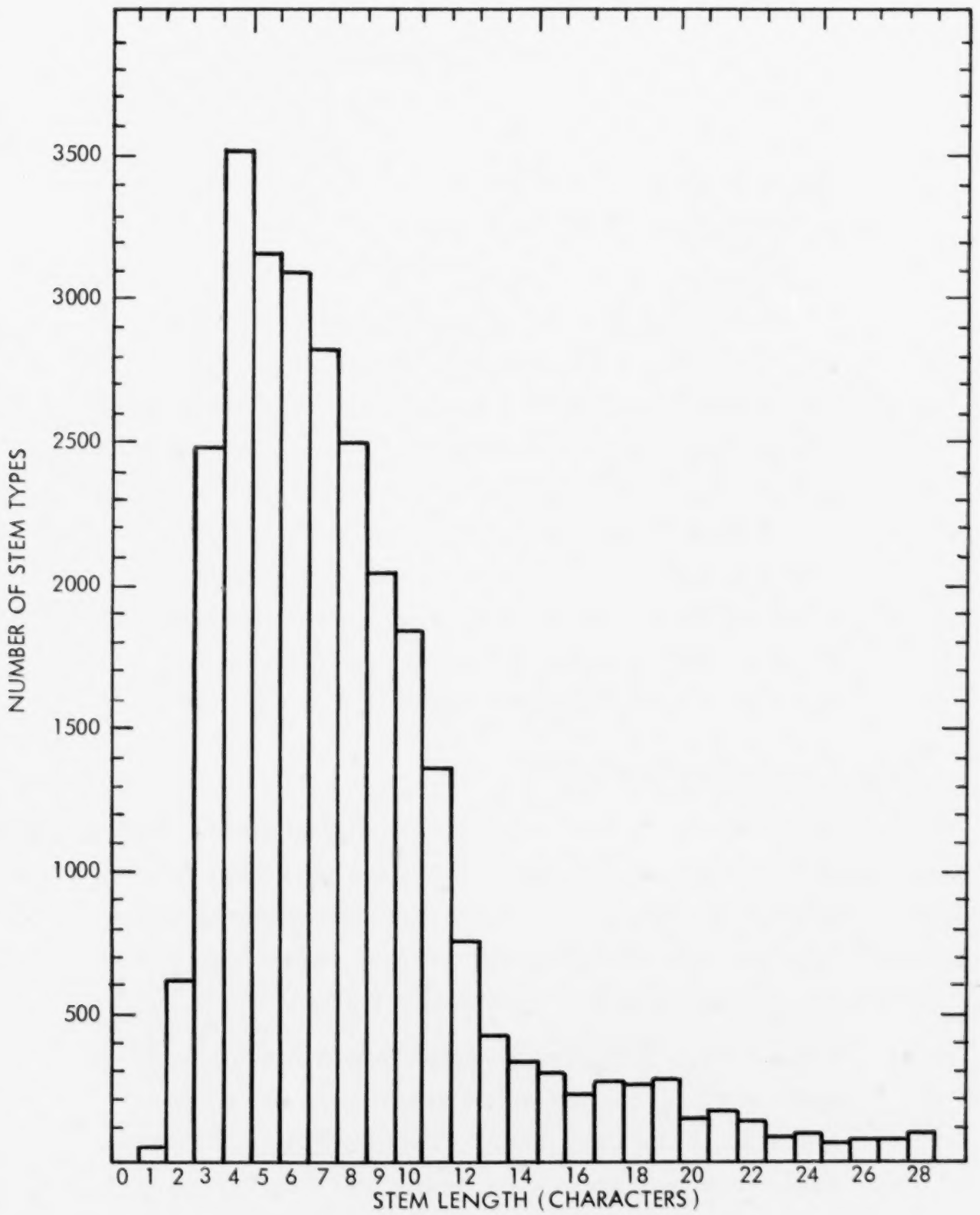

Fig. 2.4 Length Distribution of Stem Types

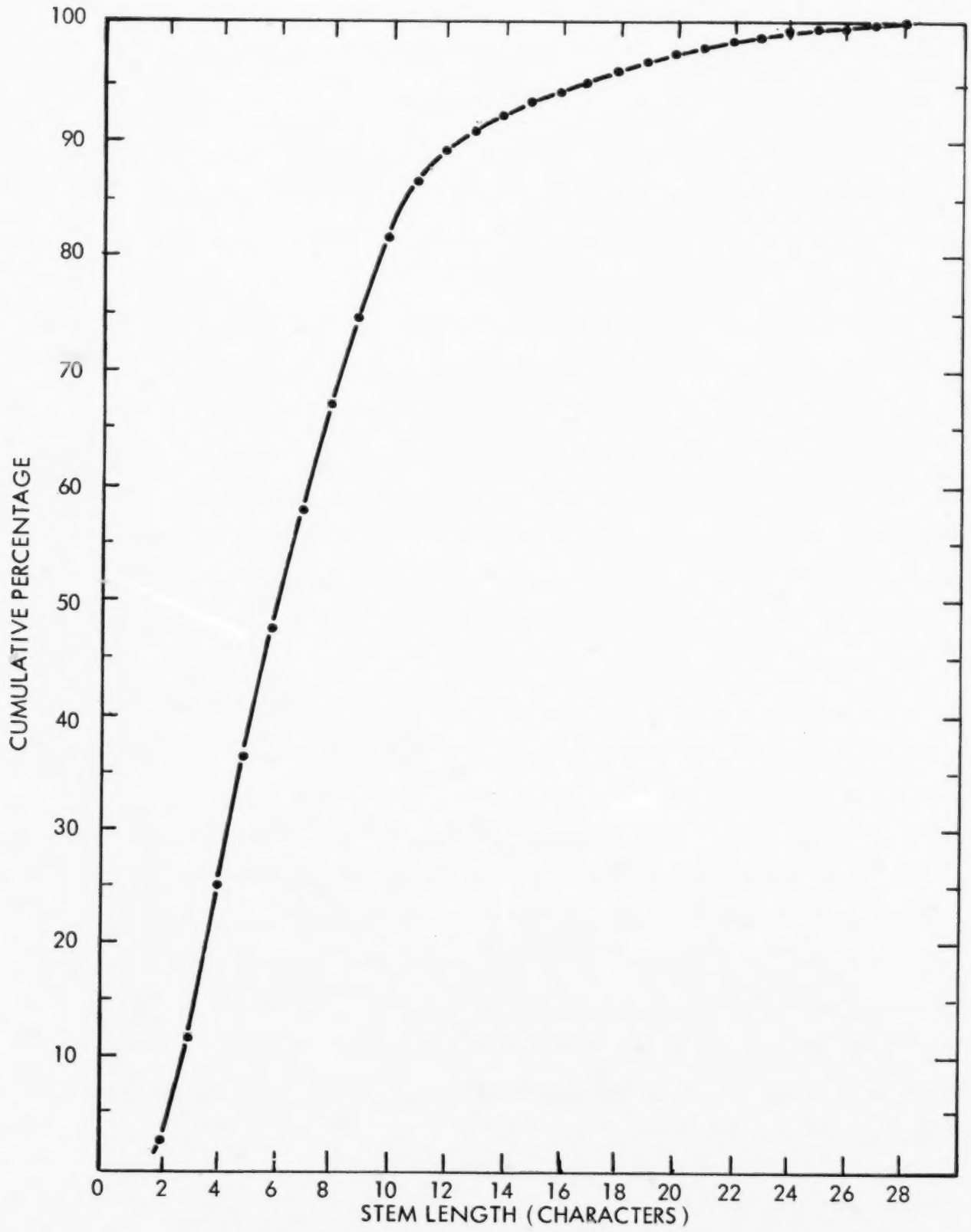

Fig. 2.5 Cumulative Percentage of Stem Types as a Function of Stem Length

(2) Special notations for character conversion[2] increase the stem type length. For example, H_2O is expressed in the inverted file as "H*sub 2*O". Many stem types of frequency 1 are of this form (see Chapter 4).

The stemming procedure is not applied to any word less than 4 characters long. Consequently, the "stem" length for length 1, 2, and 3 actually means the "word" length. There are 3149 (11.5%) such short words.

If the stem length exceeds 28, the stem is automatically rejected from the inverted file by the inverted file generation program. The number of such long stems and its distribution is unknown. As Fig. 2.4 shows, there are 90 stems with exactly 28 characters each. The distribution of longer stems could be estimated from the tail of Fig. 2.4. The specific stems with 27 or 28 characters were cursorily looked at, and these were all formulae or special symbols rather than technical words, in which case one could assume the same for stems having more than 28 characters.

There exists one stem the length of which is zero (0). This null (vacant) stem was created during the file merging process. The reason is not clear yet.

From Fig. 2.5, 68 percent of all stem types are not more than 8 characters long, 90 percent are not more than 12 characters long. From an Intrex system point of view, limiting the longest stem length to 12 (3 computer words in CTSS) or 8 (2 computer words) will neither reduce the programming effort nor increase the search efficiency by much. Neither will it release a lot of file space since the number of extremely long stems is small. But it may unburden the user's typing difficulty if the longest length be limited.

2.5 Stem Attributes

2.5.1 Introduction

Each occurrence of a stem is stored in the inverted file along with (1) the number of the document catalog record in which it occurred,

(2) attributes of the document, such as a professional journal article, (3) the word ending that the stem occurrence had, (4) word number which indicates the relative position of the stem within the index term, (5) term number, which indicates the relative position of the index term within the list of index terms for a document, and (6) a range number, which indicates the relative document length covered by the specific index term in which the stem is used. This section discusses range number, word number, and term number.

2.5.2 Range Number

An aspect of Intrex indexing is the concept of "range number".

Indexing is performed by selecting terms which describe a document in depth. A term may be a single word or a phrase (usually the latter). Each selected term is assigned a range of 0, 1, 2, 3, or 4.[10] Each range has the following meaning:

Range	Meaning
1	MAJOR subject of the document
2	SECONDARY subject of the document (a section of the document)
3	MINOR subject of the document (a fraction of the document)
4	A Laboratory TOOL or theoretical technique referred to in the document
0	A GENERIC (broader) term to the subject of the document

INTREX users can specify that only certain ranges are acceptable. Those upper-case words are command language mnemonics for this subject search specification.

In the inverted file, Range 5 is used to distinguish title words from subject index words.

Figure 2.6 shows the frequency distribution of range numbers by stem occurrences. Of all 1,204,282 stem occurrences, 33.1 percent (398,183) are used for range 2 terms, 28.9 percent for range 3.

Title words occupy a little less than 10 percent of all stem occurrences. These 112,157 stem occurrences for 15,845 document titles

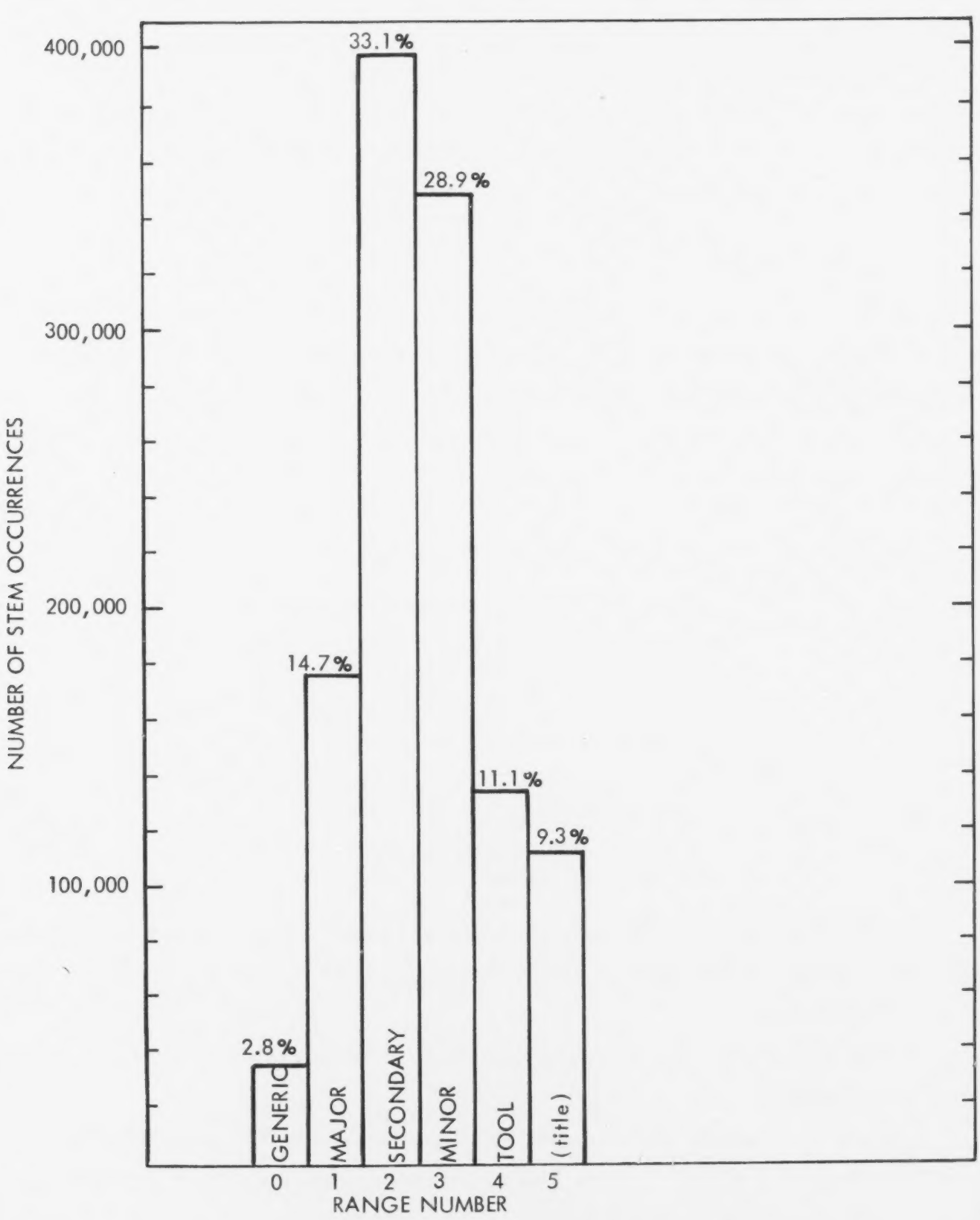

Fig. 2.6 Range Number Distribution

indicate that average length of a title is 7.1 words (excluding common words). It also indicates that if a user searches by "subject" (without any range restrictions) instead of by "title", he will get 10 times as many documents per search request. The effect of restricting subject terms by range number needs more careful examination. The restricted search:

subject magnetic/range generic

cannot broaden the retrieval ability (in the sense of retrieving an entire subject class of documents) though it may retrieve only a small part of the whole relevant set. This is because indexing practice does not assign a generic term to every document. The restricted search:

subject magnetic/range major

is meaningful in the sense that the user wants documents the major portion of which discuss "magnetic". But the restricted search:

subject magnetic/range secondary

is meaningless because the searcher is looking for documents of which only a part of each discusses the subject. The same can be said for a "minor" restriction. However a restriction on "range tool" may be useful, if it is used with proper search words which represent "tools". Thus, the "range" command is meaningful only when one of the operands is "major" or "tool." However, the other range numbers may be useful inputs to the user in his determination of the relevance of the document. He can judge if the document is only partly interesting or fully interesting by reading the index terms together with their range numbers.

2.5.3 Word Number

Each word in the index is automatically assigned a word number, which indicates the relative sequential position of the word within the index term. Word number 1 means it was the first word of an index term. Common words are culled from index terms before word numbers are determined and consequently are excluded from the word counts in each term. Note Fig. 2.7 shows word number distribution by stem occurrences on semi-log paper. Since only 4 bits are used for storing

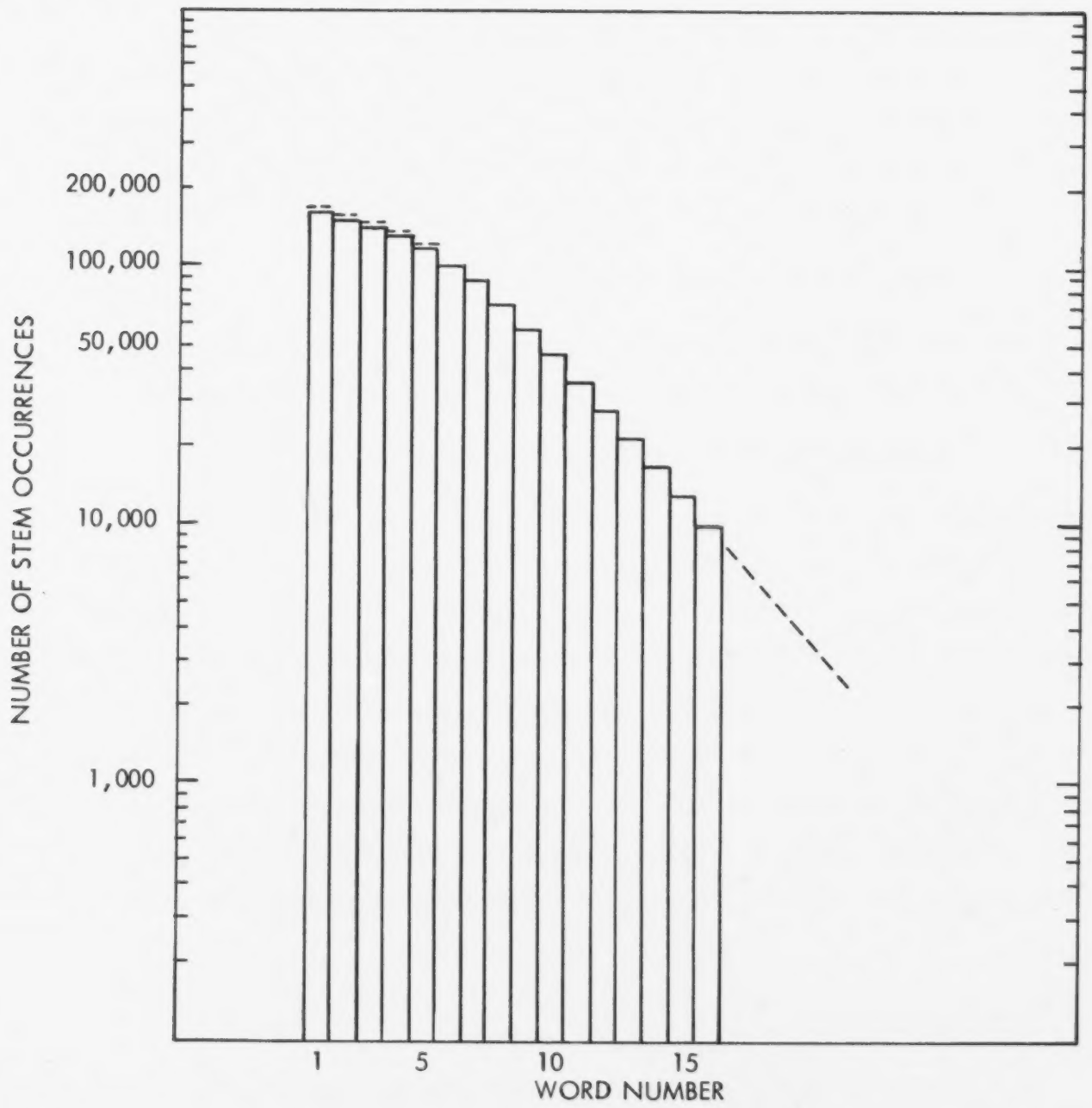

Fig. 2.7 Word Number Distribution

each word number, numbers are expressed with modulo 16. The original data (dotted line) has been adjusted to rectify this modularity (by estimating from the graph values for word numbers greater than 16). The difference between the number of stem occurrences for word number i and word number i+1 indicates the number of index terms containing i stems.

The distribution of index terms over the number of words per term is shown in Fig. 2.8. Index terms with 6 stems appear most frequently. The average number of stems per index term is 6.9 (a little less than expected).

The number of stems per index term can also be calculated in another manner because of the modulo 16 representation of word number. Suppose the number of stem occurrences with word number i is n_i. From the data above,

$$n_1 + n_{17} + n_{33} + \ldots = 165148$$

$$n_{16} + n_{32} + n_{48} + \ldots = 9808$$

$$(n_1 + n_{17} + n_{33} + \ldots) - (n_{16} + n_{32} + n_{48} + \ldots)$$

$$= n_1 - (n_{16} - n_{17}) - (n_{32} - n_{33}) - \ldots$$

$$= 165148 - 9808$$

$$= 155340$$

Then $155,340 < n_1 < 165,148$

From the graph, n_{17} is estimated as 7500, n_{33} almost zero. Thus n_1, the number of stem occurrences with word number 1, can be estimated as 157,600. Since each index term consists of at least one word, we can estimate the total number of index terms in the whole data base as 157,600. Dividing by the number of documents in the base, the average number of index terms per document turns out to be 10. By dividing 76 (the average number of subject index words per document) by 10 or dividing 1,204,282 (the total number of stem occurrences) by 157,600 (see Table 2.1), the average number of stems per index term is 7.6.

Note that the former value of 6.9 stems per term was obtained within the range of index term lengths of 1 to 15 words. If we expand the range from 1 to 20 words (using estimated values), the average becomes 7.4, which is pretty close to the 7.6 stems per index term calculated by the second method.

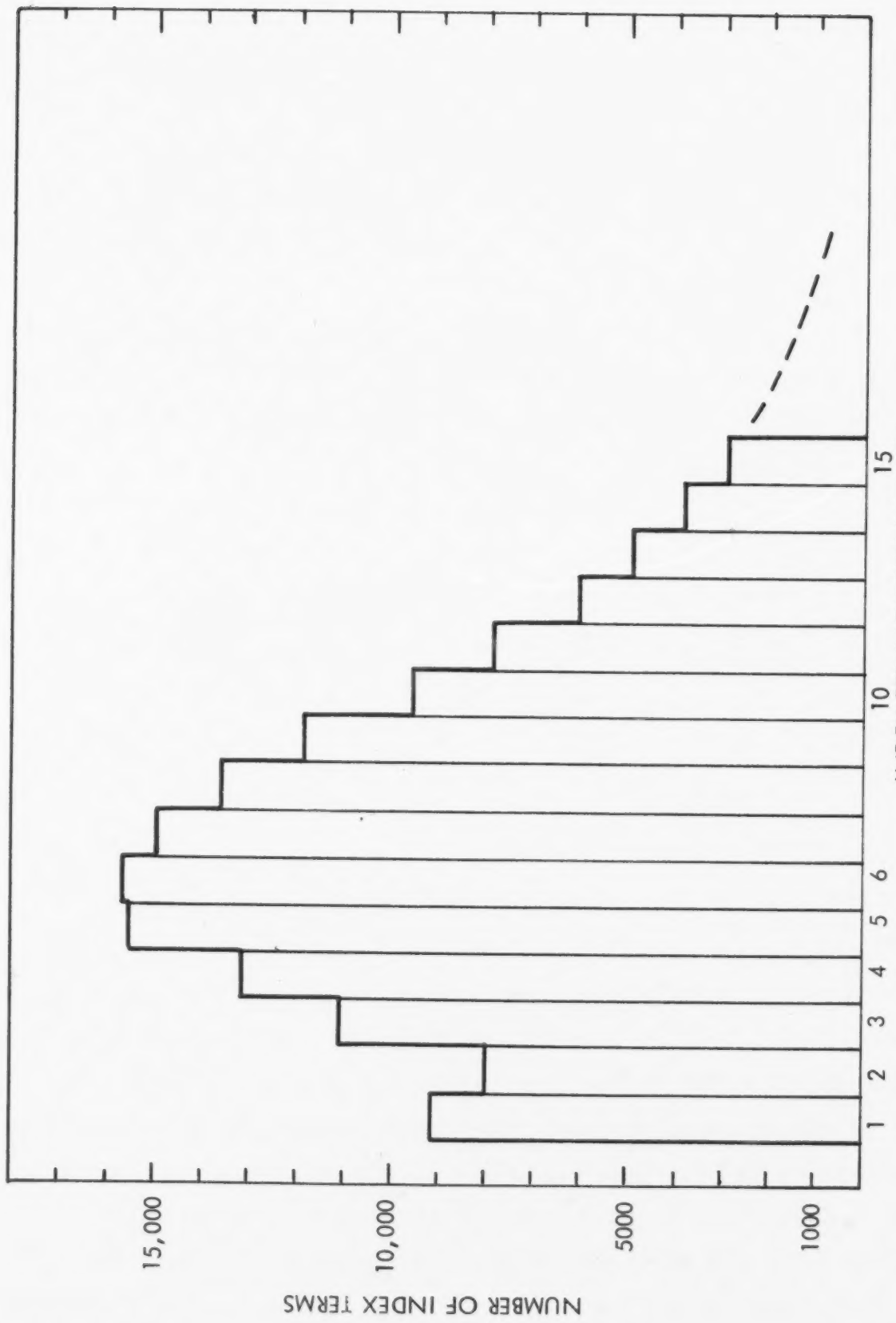

Fig. 2.8 Index Term Length Distribution

A MEDLARS report says that depth journal articles were assigned an average of 10 subject headings and non-depth journals were assigned an average of 4. Each subject heading consists of one or two (sometimes three) words. At Intrex, articles were assigned an average of 10 index terms and each term consisted of an average of 7.6 stems.

The number of common words culled out during file generation is about 25 percent of the remaining words. Cumulating this value, the number of actual words for one index term is about 9.5.

Suppose a user issued a search command "subject magnetic field". The current system searches documents with an index term which contains both "magnetic" and "field". This is done by matching term numbers (see Section 2.5.4). By using word numbers, we can further check if the stem "field" appeared right after the stem "magnet-" or not. That was an implicit purpose of recording word numbers. Four bits (modulo 16) is not quite ample for this purpose.

There appeared an irregular peak at terms with only one stem (Fig. 2.8). A number of short index terms can be found especially with range number 4. Some of them are only one word (stem) long. An extreme example, document 2412, contains 11 such single word terms all with range number 4. Such terms may have caused the irregular peak in Fig. 2.8.

2.5.4 Term Number

Each index term (a phrase containing one or more words) is automatically assigned a term number, which indicates the relative position of the index term within the list of index terms for a document. If a document is assigned more than 63 index terms, only the first 63 index terms are incorporated into the inverted file (though all terms are recorded in the catalog record file.) Term number 0 is used for titles.

Figure 2.9 shows the frequency distribution of term numbers on semi-log paper. The range number calculation (see Fig. 2.6) indicates that 112,157 stem occurrences are for titles. However the number of stem occurrences for term number 0 (i.e., titles) is 112,203. The reason for the difference (46) is under investigation.

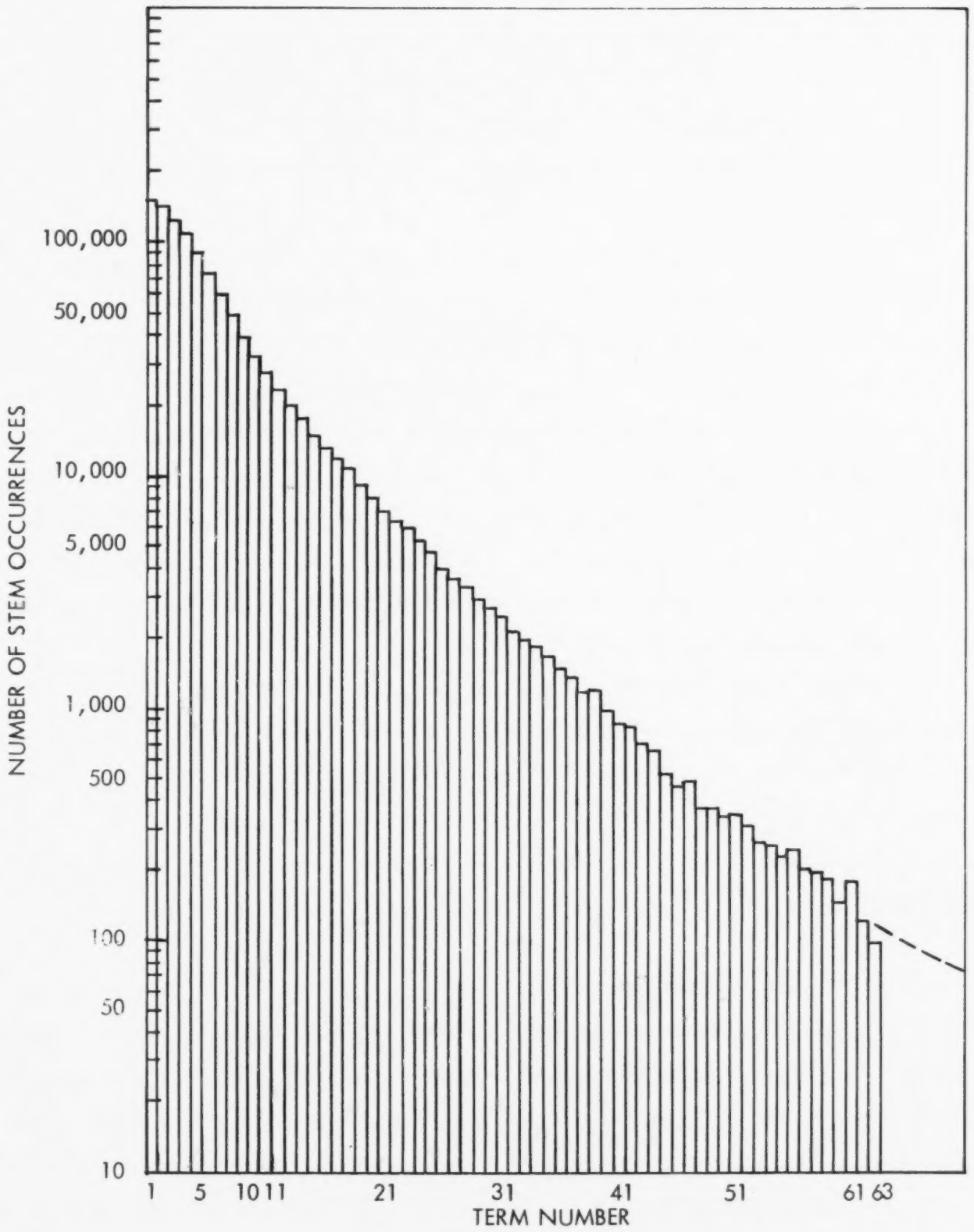

Fig. 2.9 Term Number Distribution

Another estimation of index term length is possible using term numbers. Each document is assigned at least one index term. Then there should be 15,845 index terms with term number 1. This leads to an average of 9.2 stems (with term number 1) per document. In other words, the average length of the first index term is 9.2 stems. Estimation based on word number showed the average over all terms to be 7.6 stems per term. From actual INTREX catalog file listings, it is known that the first index term in each document carries, more often than not, a range number of 1. We do not expect terms with different range numbers to have the same average term length.

By using the average of 7.6 stems for one index term, we can estimate that more than 4,000 documents were assigned 10 or more index terms each, more than 1,000 documents were assigned 20 or more. There are 10 or so documents each with more than 63 index terms. Actually the author did find a couple of these documents and they had all been processed during the earlier stages of the Intrex program.

2.6 Correspondence between Stem and Word

This section discusses an aspect of the Intrex stemming procedure that will help in a more complete evaluation in the future. The point here is focused on the degree to which a stem type can represent word types.

The current inverted file contains 35,645 word types represented by 27,291 stem types. See Table 2.2. A stem represents an average of 1.3 word types. But 84 percent of all stems (22,973) represent only one word type each. There are two cases of one-to-one correspondence between a word and a stem, either a word is stemmed or not. In the first case, "single" is stemmed to "singl-e", but the stem "singl-e" represents this word type "single" only. In the second case, words like "between", "two", "Fe", or most of the chemical formulae are never stemmed. In this case, the word remains as a word and never becomes a stem, but from the Intrex programming point of view, it is regarded as a stem.

Table 2.2

Stem Type Distribution as a Function of the Number of Word Types It Represents

Number of Endings	Number of Stem Types	Percentage (rounded)	Total Number of Word Types
1	22973	84.2	22973
2	2630	9.6	5260
3	710	2.6	2130
4	426	1.6	1704
5	236	0.9	1180
6	124	0.5	744
7	78	0.3	546
8	43	0.2	344
9	27	0.1	243
10	12	0.2	120
11	16		176
12	2		24
13	4		52
14	5		70
15	2		30
16	2		32
17	1		17
Total	27291		35645

Out of those 22,973 one-to-one corresponding stem types, 13,085 stem types occur only once in the whole indexing process. This accounts for 57 percent. The remaining 9,888 stems were used not less than twice.

If those 22,973 stem types are excluded from consideration, a stem type represents an average of 2.9 word types. As Fig. 2.10 shows, there is another extreme case. There are 5 stems each of which represents more than 15 word types. Some of them (form-, pol-) have already been mentioned in reference 12 as overstemming errors. "magnet-" would be an excellent example of the effectiveness of the stemming procedure. The relatively low frequency of the other 4 stems (compared with "magnet-") may indicate overstemming errors.

From the computer storage space point of view, the stemming procedure is not very effective. If the inverted file consisted of word types (instead of stem types), it would need at least

$$(35645 - 27291) \times 4 = 33416$$

more computer words for headers. This would cause only a little more than a 2 percent increase of storage space since the inverted file now occupies 1,481,328 computer words.

The major purpose of "stemming" is not directed to storage requirements but to morphological word control on a free vocabulary for retrieval purposes. Thus, the real evaluation of the stemming procedure should be on the basis of the stem types that actually represent more than one word type (some 16 percent of the total number of stem types).

2.7 In-Depth Indexing and Redundancy

2.7.1 Introduction

One of the core programs of Project Intrex is an augmented catalog. As a part of the catalog, subject indexing of documents is done in-depth. Statistical measures of Intrex subject indexing depth have been investigated through computer programs. Redundancy of indexing is also discussed.

For recently indexed documents, an average of 63 subject index words were used per document, and among them were 36 distinct stem types (see Fig. 2.11).*

* These values do not include the usage of 13 common words. See Chapter 1.

Stem	Frequency
diffus-	2407
diffuse	96
diffused	69
diffusely	4
diffusion	2009
diffusive	18
diffusible	1
diffusional	35
diffusivity	132
diffusing	13
diffusivities	18
diffuseness	6
diffusant	2
diffusian	1
diffuses	2
diffusively	1
form-	2357
form	607
forms	129
formal	20
formed	213
formate	13
forming	97
formates	4
formation	1139
formability	5
formalism	108
formations	10
formative	1
formic	5
formally	2
forme	1
form's	1
formable	2
magnet-	16354
magnet	236
magneto	411
magnets	299
magnetic	12036
magnetism	165
magnetize	2
magnetons	15
magnetized	216
magnetizing	24
magnetically	290
magnetization	2585
magnetizations	50
magneton	15
magnetizable	3
magnetical	1
magnetics	6

Stem	Frequency
pol-	2417
pol	16
pole	92
poly	105
polar	247
poled	11
poles	40
polarity	17
polished	50
polarizing	29
polarized	458
polarization	1295
polarizations	50
poling	4
polars	1
polation	1
polarize	1
rot-	2043
rotary	17
roton	35
rotons	11
rotated	22
rotator	32
rotating	387
rotation	877
rotatory	38
rotational	548
rotations	55
rotationally	6
rotate	3
rotates	1
rotators	10
rotable	1

Fig. 2.10 Stems that represent more than 15 word types.

(Note: The order of the full word types under each stem type is that in which they were first encountered in inverted file generation processing.)

At the initial stage of Intrex indexing, the figures were 120 words per document including 50 stem types. Redundant usage of subject index words has clearly been decreased, but the amount of information one subject index carries is decreasing too, if we measure it by the number of stem types per document.

Before going further, it should be noted that the Intrex system utilizes this redundancy. When a user issues a search command with more than one subject word ("subject temperature effect") or when he tries a Boolean operation using a "with" command, the system program checks whether the subject words are in the same index term or not. This is a loose version of phrase matching. It implies the assumption that when those two words are used in one term, they are probably used as a phrase. A certain amount of redundant word usage is necessary for this kind of phrase matching.

The Intrex indexing procedure is based upon index terms which are structured to contain sufficient context so that each term may be understood by itself. The long "stand alone" term is expected to give a user a more comprehensive understanding of the subject contents than a simple list of keywords.[3] This procedure requires a certain amount of redundant word usage, at least in the sense of word occurrences, although the context of each occurrence may be different.

A fuller discussion about the amount of unavoidable redundancy is not included here. In the following sections, the word "redundancy" refers to both avoidable and unavoidable redundancy.

2.7.2 Redundancy Measurement

Each document in the Intrex data base has its own document number and it usually reflects the approximate date the document was indexed. Documents with lower document numbers were usually indexed earlier than those with higher numbers. Since this is the only field in the inverted file that can be used as a time-varying parameter, it has been utilized for redundancy measurement.

Note that there is no precise relationship between document number and date of indexing, but only a rough one. Note also that many document numbers are not used or not yet used for various reasons. Though

the number of documents is 15,845, document numbers range from 1 to well over 20,000.

Suppose we call the document number range from 1000 (i-1)+1 through 1000 i as "document interval i". For each interval i, the number of documents, total number of subject index words, and total number of stem types taken from each document record has been calculated by computer program. The average number of subject index words per document (W_i), and the average number of stem types per document (S_i) were calculated manually for each interval i. A redundancy ratio (RR_i)

$$RR_i = W_i/S_i$$

will give a measurement for redundant word usage in the subject index.

There could be two slightly different measurements, i.e., (1) redundancy caused by repeating the same "word" type in a subject index, or (2) redundancy caused by repeating the same "stem" type in a subject index. Though both are too simple to reach any syntactic or semantic aspects of indexing, the latter measurement would be better since the current Intrex retrieval program is based upon the matching of stems. The ratio RR_i above measures the latter kind of redundancy.

2.7.3 Reduction in Redundancy

Figure 2.11 shows the graph of the number of subject index words, and the number of stem types per document in each document interval. It also shows cumulative data per document interval. Figure 2.12 is a graph of the redundancy ratio RR_i. The ratio was calculated for (1) all stems, and (2) only for high frequency stems (those stems that have more than 1000 occurrences each in the total inverted file).

A cumulative curve of the former ratio is also shown. Table 2.3 is a table of values from which the graphs are drawn. The flow of the curve can be analyzed as follows:

(1) For the first several hundred documents, the indexing method was not yet stabilized. This fact may explain why the values in document interval 1 (document numbers 1 to 1000) are atypical.

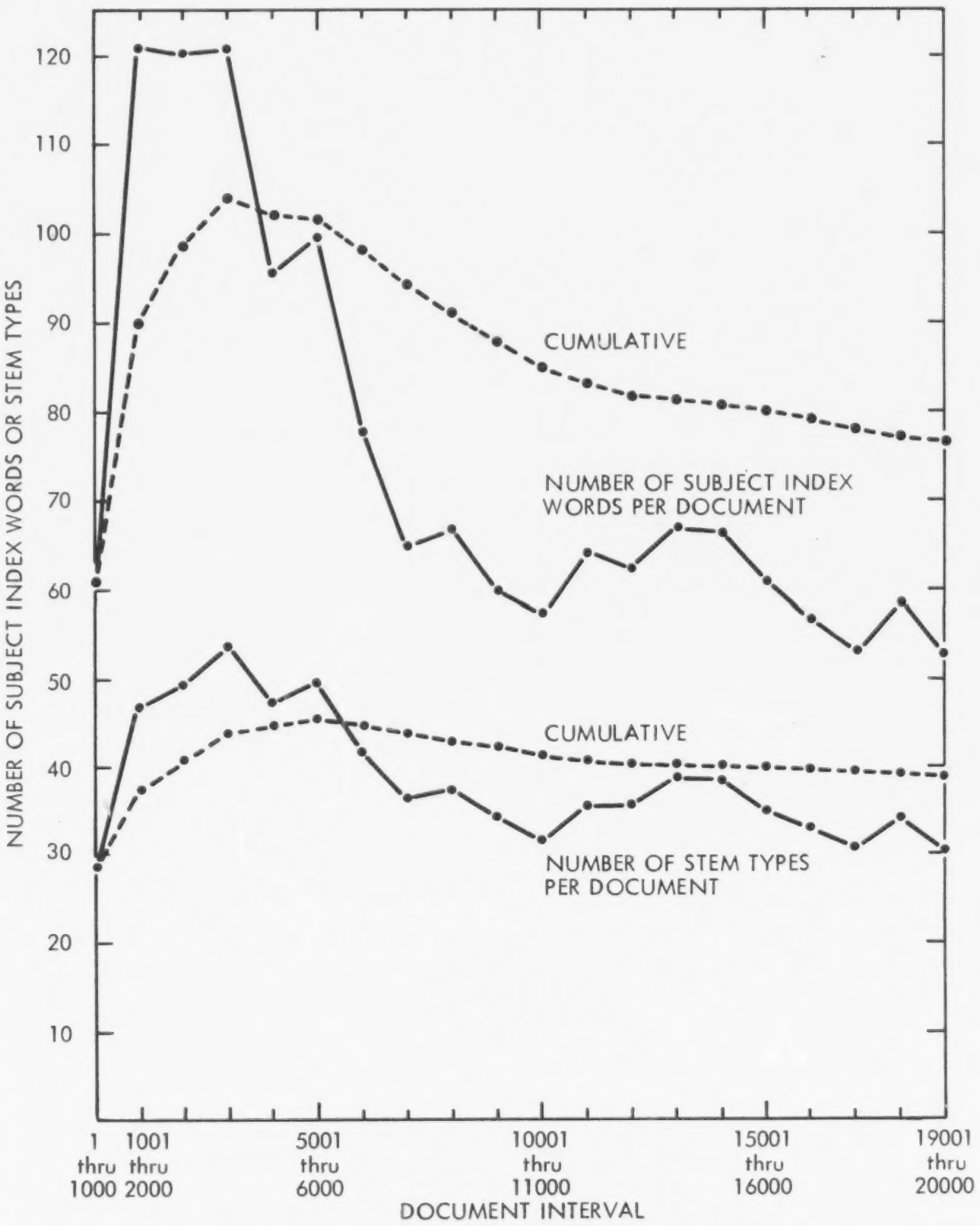

Fig. 2.11 Number of Subject Index Words or Stem Types for Each Document Interval

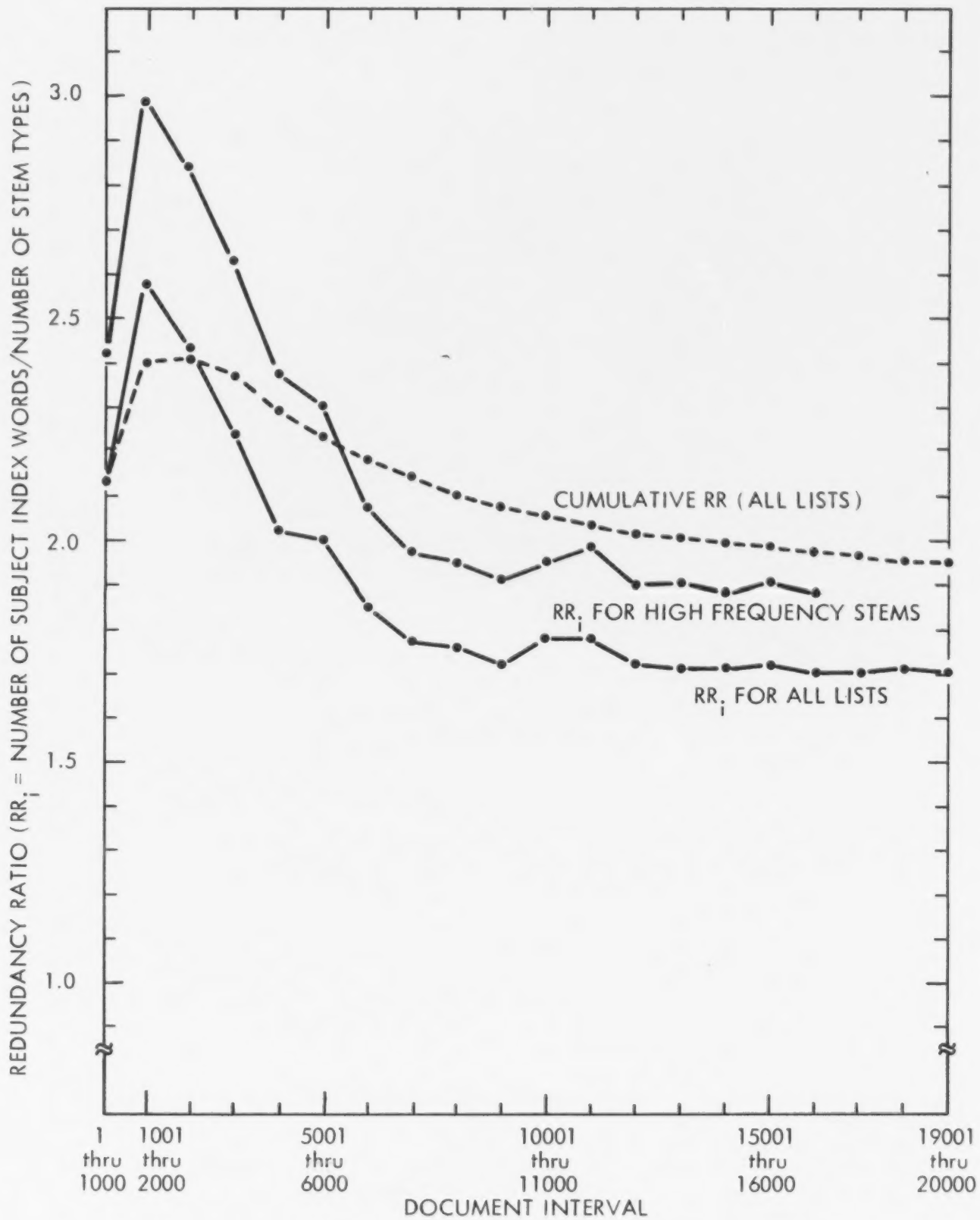

Fig. 2.12 Redundancy Ratio (RR$_i$) for Each Document Interval

Table 2.3

Number of Subject Index Words, Stem Types, and Redundancy Ratio over Document Number Intervals

Document Number	Number of Documents	Total Number of Subject Index Words	Number of Subject Index Words per Document	Total Number of Stem Types	Number of Stem Types per Document	RR
1-1000	997	60694	60.9	28476	28.6	2.13
1001-2000	930	112386	120.8	43518	46.8	2.58
2001-3000	800	95883	119.9	39417	49.3	2.43
3001-4000	823	99209	120.5	44134	53.6	2.24
4001-5000	995	95163	95.6	46942	47.2	2.02
5001-6000	961	95621	99.5	47678	49.6	2.00
6001-7000	934	72686	77.8	39105	41.9	1.85
7001-8000	891	57789	64.9	32552	36.5	1.77
8001-9000	968	64722	66.9	36626	37.8	1.76
9001-10000	982	58685	59.8	33997	34.6	1.72
10001-11000	922	52692	57.1	29482	32.0	1.78
11001-12000	914	58650	64.2	32918	36.0	1.78
12001-13000	926	57464	62.1	33276	35.9	1.72
13001-14000	477	31869	66.8	18597	39.0	1.71
14001-15000	515	34156	66.3	19962	38.8	1.71
15001-16000	476	28972	60.9	16809	35.3	1.72
16001-17000	494	27981	56.6	16404	33.2	1.70
17001-18000	629	33316	53.0	19587	31.1	1.70
18001-19000	689	40531	58.8	23630	34.3	1.71
19001-2000	350	18268	52.2	10715	30.6	1.70
20001-21000	82	4197	51.2	2495	30.4	1.68
25001-26000	90	3348	37.2	2578	28.6	1.29

(2) For document intervals 2-4 (document number 1001 to 4000), an average of 120 subject index words were used per document, and among them were 50 stem types. The average RR in those intervals is 2.42. In other words, a stem type is used more than twice in the index term set for each document in that interval.

(3) A major change in the indexing principles came in April 1968, when the first 4200 documents had already been indexed. The change can be summarized as modifications which reduced the extent of word redundancy.[10] For example, three terms "barium titanate (4)", "optical dispersion curves for barium titanate (3)", "optical dispersion curves for strontium titanate (3)", would become a single term "optical dispersion curves for barium titanate and strontium titanate (3)". Other redundancy reduction procedures are discussed in reference 10.

The effect of this change can clearly be observed in the graphs. Note again that the document numbers do not exactly reflect the time and order of the indexed documents. When 4200 documents had been indexed, document numbers ranged from 1 to 6000 or so. Some documents which have document numbers less than 4200 were indexed after the change in indexing method. The gradually decreasing curve of both the number of subject index words and the number of stem types is attributed to that.

(4) For document intervals 14 to 20 (13001 to 20000), there are many "vacant" document numbers representing documents not fully processed into the system. As is shown in Table 2.3, only about half the documents in those interval numbers have

been fully processed. Values for document intervals 21 and 26 are not shown on the graphs. Only 82 documents have been indexed in interval 21, and 90 documents in interval 26. Another reason for eliminating the interval 26 is that this range of document numbers is reserved for a different subject field of the data base (computer graphics), and the indexing procedure is quite different for that group of documents.[13] Numeric values are shown in Table 2.3 only for comparison.

Tracing the graph, it is found that the indexing depth averages vary significantly over the time period of data base build up. Further comparison was made in Table 2.4 between two stages of indexing method, the earlier stage for document intervals 1 to 6 (document numbers 1 to 6000), the later stage for intervals 7 to 20 (numbers 6,001 to 20,000). The number of subject index words has decreased 38 percent (or by 39 words per document), while the number of stem types has decreased only 21 percent (or by 10 stem types per document). The differences between those two parameters caused the reduction in redundancy. The RR (Redundancy Ratio) for recently indexed documents has dropped to 1.75. The RR for the whole inverted file has dropped to 1.95 from its former value 2.4, which shows that the purpose of changing the indexing method has been achieved.

On the other hand, the reduction of the number of stem types should be noticed carefully. In the current Intrex search procedure, the total amount of information a subject index carries is mostly determined by the number of stem types in a subject index. The 21 percent decrease in the number of stem types can result in almost the same amount of information loss; i.e., the effective depth of indexing is being reduced. Since the loss depends on the actual stem types reduced, further investigation is needed on this point. It looks like an uninvited byproduct of the change in indexing method.

Table 2.4

Comparison between Two Stages of Indexing

	Earlier Stage: Document Numbers 1 - 6000	Later Stage: Document Numbers 6001 - 20000	Percent Decrease from Earlier Stage
Number of Documents	5506	10167	
Average Number of Subject Index Words per Document	101.5	62.7	38
Average Number of Stem Types per Document	45.4	35.8	21
Redundancy Ratio (RR)	2.23	1.75	

Another interesting feature in Fig. 2.12 is that the RR for high frequency stems is usually higher than that for all lists (stem types), though the trends are alike. This means the stems are frequently used not only in many subject indices for many documents, but also within a subject index for a single document. See Section 3.3.3 for further study on this redundancy of high frequency stems.

3. STEMS WITH HIGH FREQUENCY

3.1 Introduction

In the current inverted file, there are 260 stem types with frequency more than 1000. They amount to less than one percent of the total stem types but they occupy more than 50 percent of the whole file (storage area). See Table 3.1 and 2.1.

The characteristics of these high frequency stems are investigated from various points of view, including their comparative frequency, redundant usage, and retrieving ability.

Table 3.1

Frequently-Used Stems

	Frequency More Than 1000	All	Ratio
Number of Stem Types	260	27,291	0.95%
Total Frequency (Total Number of Occurrences)	650,333	1,204,282	54 %

3.2 High Frequency Stems

Table 3.2 is the list of frequently-used stems in descending order of frequency. The most frequent stem is "magnet-" followed by "temperatur-", and then "effect-". The frequency of stems used less than 2000 times each is not shown in Table 3.2 since their order is not critical.

These high frequency stems can be categorized into at least three groups:

(1) Pure technical words: magnet-, temperatur-, cryst-, spin-, electron-, alloy-, etc.;

(2) Words sometimes used technically (or technical words that have broader meanings):
effect-, field-, funct-, depend-, system-, model-, etc.;

(3) Common words:

between, an, two, due, etc.

Groups (1) and (2) could be divided into several groups depending upon how "technical" the stem is. The main difference between groups (1) and (2) is that stems in group (1) can be meaningfully used as a single search term; on the other hand, those in group (2) would be used as a part of a search phrase that consists of more than one word. For example, a user may issue

subject magnetic
subject alloys
subject magnetic field
subject spin effect

but seldom

subject field
or subject effect.

The reader should carefully note that Table 3.2 (and similar tables given later) are based on the "stem". We can say "electron-" appeared more frequently than "alloy-". But it does not necessarily mean the word "alloy" appeared more frequently than "electronics".

Also because of the over- or under-stemming problem, the words "alloy" and "alloys" become two different stems "allo-" or "alloy-" respectively. If they were stemmed properly, it would put the stem 6th (between "field-" and "funct-").

Another point to be mentioned again is that no vocabulary control has been incorporated in the indexing process. In Table 3.2 similar concepts appear in several positions. For example, "iron-" (19th), "Fe" (58th) and maybe "steel" (60 th). or "nickel-" (66th) and "Ni" (89th), or "alumin-" (93rd) and "Al" (110th), or "therm-" (74th) and "heat-" (76th). A comparative statement such as "'Al' is used less frequently than 'Ni'" is meaningless if it refers to the concept and not to the stem alone.

Table 3.2

List of High Frequency Stems in Descending Order of Frequency
(15, 845 Document Collection)

	Stem	Frequency		Stem	Frequency
1	magnet-	16824	23	laser-	5197
2	temperatur-	13502	24	met-	5164
3	effect-	12883	25	phas-	5133
4	cryst-	11327	26	propert-	4936
5	field-	10260	27	liquid-	4903
6	funct-	7632	28	opt-	4810
7	wav-	7374	29	frequ-	4809
8	electron-	7282	30	ferromagnet-	4714
9	spin-	6971	31	spectr-	4569
10	depend-	6682	32	interact-	4564
11	alloy-	6625	33	singl-e	4502
12	energ-	6228	34	between	4244
13	reson-	5844	35	equ-	4182
14	scatter-	5752	36	high-	4032
15	stat-	5720	37	ion-	3878
16	transit-	5699	38	an	3701
17	theor-	5551	39	const-	3532
18	structur-	5423	40	gas-	3514
19	iron-	5406	41	surfac-	3452
20	system-	5372	42	light-	3402
21	film-	5361	43	order-	3402
22	model-	5283	44	stress-	3368

Table 3.2 (Cont'd)

Stem	Frequency	Stem	Frequency
45 solut-	3326	69 measur-	2795
46 electr-	3324	70 coeffici-	2790
47 solid-	3309	71 excit-	2780
48 anisotrop-	3331	72 typ-	2676
49 calcul-	3220	73 using-	2640
50 tim-	3220	74 therm-	2602
51 lattic-	3219	75 lin-	2583
52 two	3192	76 heat-	2569
53 allo-	3186	77 crit-	2546
54 rel-	3166	78 diffus-	2516
55 mod-	3164	79 phonon-	2503
56 absorpt-	3101	80 pol-	2503
57 dens-	3062	81 conduct-	2494
58 Fe	3056	82 disloc-	2485
59 mechan-	3053	83 rot-	2468
60 steel-	3030	84 form-	2464
61 ferrit-	3019	85 distribut-	2445
62 nucle-	3002	86 mater-	2442
63 low-	2941	87 antiferromagnet-	2439
64 molecul-	2874	88 measurem-ents	2419
65 relax-	2848	89 Ni	2419
66 nickel-	2837	90 elast-	2375
67 atom-	2825	91 point-	2362
68 method-	2800	92 band-	2356

Table 3.2 (Cont'd)

	Stem	Frequency
93	alumin-	2310
94	determin-	2279
95	techniqu-	2254
96	growth-	2253
97	stud-	2213
98	pressur-	2207
99	*deg*K	2179
100	bound-	2163
101	domain-	2161
102	oscill-	2157
103	strain-	2148
104	coupl-	2142
105	rot-	2130
106	behavior-	2114
107	semiconduct-	2089
108	puls-	2059
109	impur-	2052
110	particl-	2051
111	Al	2050
112	process-	2044
113	thin	2022
114	composit-	2016
115	experim-	2009

(Each of the following stems has 2,000 to 1,000 frequency. They are arranged in the descending order of frequency, but frequency is omitted.)

116 dielectr-	135 appl-
117 Raman-	136 paramagnet-
118 garnet-	137 chang-
119 strength-	138 direct-
120 parameter-	139 suscept-
121 express-	140 copper-
122 radi-	141 correl-
123 intens-	142 gener-
124 oxid-	143 concentr-
125 veloc-	144 *percent*
126 second-	145 beam-
127 near-	146 cub-
128 var-	147 dispers-
129 vari-	148 used
130 grain-	149 character-
131 due	150 differ-
132 resist-	151 moment-
133 exchang-	152 bas-
134 rar-e	153 earth-

Table 3.2 (Cont'd)

154 dur-	179 diffract-	204 wall-
155 X	180 ultrason-	205 microscop-
156 fre-	181 pot-	206 dat-
157 sound-	182 observ-	207 fluctu-
158 supercond-	183 valu-	208 ray-
159 power-	184 shift-	209 dimens-
160 rang-	185 part-	210 siz-
161 or	186 pur-	211 fact-
162 helium	187 reg-	212 curv-
163 curr-	188 vibr-	213 forc-
164 Cu	189 attenu-	214 modul-
165 deform-	190 induc-	215 Curi-e
166 level-	191 mot-	216 nonlinear-
167 acoust-	192 dop-	217 cobalt-
168 propag-	193 N	218 compound-
169 approxim-	194 linear-	219 dilut-
170 analys-	195 term-	220 maser-
171 ferroelectr-	196 axi-	221 fatigu-
172 creep-	197 silicon-	222 *deg*C
173 cast-	198 hard-	223 The-
174 microwav-	199 carbon-	224 flow-
175 center-	200 Co-	225 yttrium-
176 crack-	201 condit-	226 fractur-
177 plan-	202 comparison-	227 emiss-
178 neutron-	203 dynam-	228 Permallo-y

Table 3.2 (Cont'd)

229 influ-	245 glass-
230 one-	246 YIG
231 compon-	247 longitudin-
232 dipol-	248 loss-
233 rub-	249 infrar-ed
234 reflect-	250 cross-
235 polycrystall-	251 satur-
236 amplitud-	252 kinet-
237 test-	253 sign-
238 mixtur-	254 Cr
239 melt-	255 interfac-
240 oper-	256 transform-
241 spher-	257 sampl-
242 transvers-	258 hydr-
243 compar-	259 shear
244 chromium	260 under

Table 3.2 should be utilized to check important stem synonyms; without them the user may lose large amounts of information.

A search using only one of these stems will produce a list of a thousand or more documents. Intrex instantly indicates to the user through a feedback message, the number of retrieved documents. If this is too large a set, the user is then able to recover by narrowing the search scope. In this sense there are no harmful problems associated with these high frequency stems. In some search situations, particularly those where true synonyms are involved, a user may require an even larger base. For example, if a user issued the command

"subject nickel",

he should be advised automatically to incorporate another subject term "Ni", such as in the message: "Intrex recommends that you incorporate the following words in your search -- Ni". Thus, a potential significant loss in basic scope could be recovered. It should not be too difficult to organize a thesaurus for only these high frequency stems.

3.3 Change in High Frequency Stem Order

Table 3.2 showed the status of high frequency stems. The next question is how this status or order changed during successive file updates.

The inverted file listing for the 955 document collection was examined and those stems, each of which had a frequency greater than 200, were marked. Table 3.3 is a list of those 90 stems in descending order of frequency. Comparing Table 3.2 with Table 3.3, we can conclude that those 90 stems continued to be high frequency stems through the most recent version (May 1971) of the inverted file. Of those 90 stems, 85 now have a frequency greater than 2000 each and they can be found within the first 115 high frequency stems in Table 3.2.

There are a couple of extreme examples. A group of stems used very frequently when the document collection was 955 decreased in relative frequency after that. Though the document collection increased more than 15 times (955 → 15848), the frequency of these stems increased by factors of only 6 to 9. Examples are: "*percent*" (216 → 1723), "alumin-" (254 → 2310), "helium-" (287 → 1602). On the other hand, stems "electr-" (199 → 3324), "relax-" (135 → 2848), "stress-"

Table 3.3

List of High Frequency Stems in Descending Order of Frequency
(955 Document Collection)

	Stem	Frequency		Stem	Frequency
1	magnet-	1533	24	scatter-	459
2	temperatur-	1145	25	ferromagnet-	440
3	effect-	1130	26	alloy-	439
4	field-	809	27	phas-	436
5	cryst-	797	28	equ-	425
6	transit-	688	29	spectr-	416
7	funct-	649	30	frequ-	407
8	spin-	603	31	gas-	396
9	reson-	590	32	interact-	395
10	wav-	579	33	nickel-	393
11	liquid-	567	34	order-	377
12	system-	562	35	solid-	375
13	depend-	558	36	met-	373
14	laser-	552	37	ion-	367
15	film-	548	38	high-	363
16	stat-	548	39	singl-e	350
17	model-	529	40	steel-	349
18	electron-	528	41	lattic-	340
19	theor-	519	42	opt-	329
20	energ-	507	43	const-	324
21	structur-	506	44	calcul-	316
22	iron-	505	45	between	312
23	propert-	478	46	two	306

Table 3.3 (Cont'd)

	Stem	Frequency		Stem	Frequency
47	anisotrop-	304	71	rot-	235
48	absorpt-	303	72	lin-	233
49	solut-	300	73	rel-	232
50	molecul-	297	74	allo-	231
51	crit-	292	75	antiferromagnet-	228
52	helium	287	76	heat-	225
53	Fe	281	77	superconduct-	222
53	nucle-	281	78	garnet-	220
55	excit-	259	79	*percent*	216
56	an	258	80	method-	213
56	pol-	258	81	express-	210
58	typ-	256	82	phonon-	209
59	atom-	254	82	thin-	209
59	alumin-	254	84	mod-	206
61	ferrit-	250	85	tim-	205
62	dens-	249	85	distribut-	205
63	low-	248	85	coupl-	205
64	coeffici-	247	88	elast-	203
65	light-	244	89	oscill-	201
66	techniqu-	240	89	Ni	201
67	therm-	239			
68	measurem-ents	238			
69	mechan-	236			
69	measur-	236			

(147 → 3368), "surfac-" (192 → 3452) increased in frequency by factors of 17 to 23 as the data base grew. (These stems are not in Table 3.3, but only in Table 3.2.) But as a whole, no significant change in order can be observed and this is further verified by Table 3.4.

Table 3.4 shows the order of the first 40 high frequency stems at several stages of the document collection size. No significant change in order is observed from this table either. The only changes with any significance are those in going from the 955 to the 4958 document collection. The earlier base was less balanced than the others because not all of the data base subject areas had been settled upon at that time.

If one wants to try to organize a file that treats those high frequency stems (which require large storage space) by a method separate from that for less frequent stems, one can predict that the current high frequency stems will continue to remain high frequency stems unless the data base scope is changed drastically.

Table 3.4

Frequency Order at Each Stage of Document Collections
(The numbers indicate the rank order of frequency.
The stems are listed by rank order for the 15,845 document collection)

	15845	14256	11326	10200	9298	7593	5948	4958	955
magnet-	1	1	1	1	1	1	1	1	1
temperatur-	2	2	2	2	3	2	2	3	2
effect-	3	3	3	3	2	3	3	2	3
cryst-	4	4	4	4	4	4	5	4	5
field-	5	5	5	5	5	5	4	5	4
funct-	6	6	6	7	7	8	8	9	7
wav-	7	7	8	8	9	9	9	8	10
electron-	8	8	9	9	8	7	7	7	18
spin-	9	9	7	6	6	6	6	6	8
depend-	10	11	11	11	12	14	14	14	13
alloy-	11	10	10	10	10	10	12	12	26

Table 3.4 (Cont'd)

	15845	14256	11326	10200	9298	7593	5948	4958	955
energ-	12	12	12	12	15	15	16	15	20
reson-	13	13	13	15	14	12	11	11	9
scatter-	14	16	24	23	23	23	23	21	24
stat-	15	14	16	16	16	17	17	17	16
transit-	16	15	18	18	18	16	14	13	6
theor-	17	17	17	17	17	18	18	18	19
structur-	18	20	19	19	19	19	21	20	21
iron-	19	18	14	13	13	13	13	16	22
system-	20	21	20	20	20	20	19	19	12
film-	21	19	15	14	11	11	10	10	15
model-	22	22	23	24	24	25	25	25	17
laser-	23	23	22	22	21	21	22	22	14
met-	24	24	21	21	22	24	24	26	36
phas-	25	25	27	28	28	29	31	32	27
propert-	26	26	26	26	26	26	27	26	23
liquid-	27	27	28	31	33	32	30	29	11
opt-	28	30	31	30	32	31	34	36	42
frequ-	29	28	29	29	31	30	32	34	30
ferromagnet-	30	29	25	25	25	22	20	24	25
spectr-	31	31	33	33	29	28	28	30	29
interact-	32	32	30	27	27	27	26	23	32
singl-e	33	33	32	32	30	33	33	33	39
between	34	35	35	35	37	38	38	39	45
equ-	35	34	34	34	34	35	35	31	28

Table 3.4 (Cont'd)

	15845	14256	11326	10200	9298	7593	5948	4958	955
high-	36	36	38	38	38	39	39	38	38
ion-	37	37	37	37	35	34	29	28	37
an	38	38	36	36	36	37	40	44	56
const-	39	40	41	42	42	42	44	43	43
gas-	40	39	39	40	40	40	37	35	31

3.4 Redundancy of High Frequency Stems

3.4.1 The Number of Distinct Document References

The frequency of the stem "magnet-" is 16, 824, but Intrex will retrieve only 4, 555 documents for "subject magnetic". This can be restated as "The stem "magnet-" has been used 16, 824 times during the whole indexing process, but the stem appears an average 4 times in the subject index to each of 4555 distinct documents." This redundant usage of subject words has been discussed in Section 2.7 on the basis of each document index. The redundancy of each high frequency stem is discussed here. Table 3.5 lists stems in descending order of the number of their distinct document references. The list includes only those 76 stems with more than 1000 distinct references each. Comparing Table 3.5 with Table 3.2, a considerable change is observed in the stem order, especially for technical word stems. The stem "magnet-" now ranks third in Table 3.5 instead of first, "cryst-" ranks 7th instead of 4th, "spin-" appears 26th instead of 9th, and "alloy-" went from 11th to 48th.

Stems of technical words now generally rank lower in terms of distinct document references while the stems of the less technical and the common words rose higher in rank. These changes are interpreted to mean that in those documents in which high frequency technical words are used in indexing, these words are used far more redundantly than is apparent from data for redundancy across the entire collection. Whether

that redundancy is avoidable or unavoidable (that is, necessary for the context of a stand-alone index term) has not been investigated. The less technical and common words will appear more frequently throughout the collection no matter what the technical subject of the document is, and so they would be expected to have a greater number of distinct document references than the more technical word stems. These points are examined further in Sections 3.3.2, and 3.3.3.

Table 3.5

Stems in Descending Order of the Number of Distinct Document References (15,845 Document Collection)

1	effect-	6083	18	rel-	2308
2	temperatur-	5877	19	transit-	2307
3	magnet-	4555	20	equ-	2302
4	funct-	4210	21	interact-	2292
5	field-	4172	22	system-	2257
6	depend-	4026	23	an	2212
7	cryst-	3581	24	phas-	2195
8	energ-	3023	25	high-	2166
9	theor-	2987	26	spin-	2164
10	between	2954	27	singl-e	2099
11	electron-	2907	28	spectr-	2023
12	model-	2706	29	calcul-	2020
13	propert-	2680	30	reson-	1980
14	stat-	2657	31	method-	1950
15	structur-	2561	32	const-	1914
16	wav-	2413	33	mechan-	1873
17	frequ-	2352	34	tim-	1868

Table 3.5 (Cont'd)

35	measur-	1856	
36	opt-	1810	
37	low-	1753	
38	met-	1730	
39	two	1728	
40	order-	1682	
41	scatter-	1673	
42	dens-	1620	
43	techniqu-	1616	
44	electr-	1581	
45	measurem-ents	1560	
46	coeffici-	1505	
47	ferromagnet-	1499	
48	alloy-	1495	
49	typ-	1490	
50	ion-	1433	
51	distribut-	1408	
52	liquid-	1393	
53	lattic-	1387	
54	surfac-	1386	
55	iron-	1378	
56	solut-	1374	
57	express-	1357	
58	anisotrop-	1343	
59	mod-	1332	
60	absorpt-	1290	
61	stress-	1267	
62	light-	1254	
63	laser-	1252	
64	solid-	1249	
65	therm-	1244	
66	lin-	1217	
67	heat-	1207	
68	pol-	1203	
69	atom-	1193	
70	excit-	1167	
71	coupl-	1132	
72	gas-	1110	
73	allo-	1098	
74	relax-	1059	
75	crit-	1052	
76	nucle-	1034	

3.4.2 Redundant Usage and Word Meanings

Here, we define another measurement for redundancy. Suppose a stem has been used in n indices (i.e., for n documents). Because of repetitive usage, the total frequency of the stem would become m ($m \geq n$). The Redundancy Ratio for the stem is defined as

$$RR_s = m/n$$

or RR_s = Frequency of the Stem/Number of Distinct Document References

RR_s was calculated for all stems listed in Table 3.2. The values range from 5.52 to 1.01. Instead of ordering them by RR_s, they are grouped by ranges of RR_s (from 5.9 to 5.0, 4.9 to 4.0, ...) and arranged into alphabetical order within the range. The result is shown in Table 3.6. The stem with the most redundant usage is "film-" (5.52), followed by "maser-" and "steel-". The stem with the least redundancy is "The-" (1.01; stem "The-" represents words "The" and "They").

Among the interesting features of Table 3.6 are:

(1) Technical words have highly redundant usage ($RR_s > 2.0$ generally).
(2) The lower the RR_s becomes, the more common the stems appear.
(3) Synonyms have similar values of RR_s.

Stems with RR_s more than 4.0 are all technical words. Most popular element or material names appear in the ranges above 3.0. In the ranges below 2.9, broader stems or concepts like "band-", "bound-", "correl-" or "effect-" appear. In the range below 1.9, stems become more common including "behavior-", "calcul-" or "compar-". Their range also includes a couple of the most common stems "an", "between", "or" or "under".

It is not proper to discuss these common word stems with the redundancy measure above. The usage of words such as "an", "between", and "under" can hardly be controlled by indexers creating a natural language, stand-alone, and readable expression. However the redundancy of more technical words may not be as controllable as one might think

Table 3.6

High Frequency Stems Grouped by the Redundancy Ratio RR_s

RR_s: 5.9 - 5.0	dilut-	spin-	copper-
film-	disloc-	wall-	crit-
maser-	domain-	wav-	cub-
steel-	earth-	YIG	curr-
RR_s: 4.9 - 4.0	Fe	yttrium-	deform-
alloy-	ferroelectr-	RR_s: 2.9 - 2.0	dielectr-
cast-	ferromagnet-	*percent*	diffus-
creep-	fractur-	absorpt-	dipol-
fatigu-	garnet-	acoust-	dop-
ferrit-	gas-	allo-	effect-
laser-	iron-	anisotrop-	elast-
Permallo-y	liquid-	antiferromagnet-	electr-
superconduct-	magnet-	atom-	electron-
RR_s: 3.9 - 3.0	mixtur-	attenu-	emiss-
Al	Ni	band-	energ-
alumin-	nickel-	beam-	exchang-
chromium	oxid-	bound-	excit-
Co-	Raman-	carbon-	field-
cobalt-	rar-e	center-	flow-
Cr	rub-	composit-	fluctu-
crack-	scatter-	compound-	frequ-
cryst-	semiconduct-	correl-	glass-
Cu	silicon-	conduct-	grain-

Table 3.6 (Cont'd)

growth-	paramagnet-	therm-	coupl-
hard-	particl-	thin-	cross-
heat-	phas-	transit-	Curi-e
helium-	phonon-	ultrason-	curv-
hydr-	pol-	RR_s: 1.9 - 1.0	dat-
impur-	polycrystall-	*deg*C	dens-
infrar-ed	pressur-	*deg*K	depend-
interfac-	puls-	amplitud-	determin-
ion-	radi-	an	diffract-
lattic-	relax-	analys-	differ-
light-	reson-	appl-	dimens-
lin-	rot-	approxim-	direct-
mater-	singl-e	axi-	dispers-
melt-	solid-	bas-	distribut-
met-	solut-	behavior-	due
microwav-	sound-	between	dur-
mod-	spectr-	calcul-	dynam-
modul-	spher-	Chang-	experim-
molecul-	stat-	character-	express-
N	strain-	coeffici-	equ-
neutron-	strength-	compar-	fact-
nonlinear-	stress-	comparison-	forc-
nucle-	structur-	compon-	form
opt-	surfac-	concentr-	fre-
order-	system-	condit-	funct-
oscill-	temperatur-	const-	gener-

Table 3.6 (Cont'd)

high	near-	ray-	theor-
induc-	mot-	reflect-	tim-
influ-	observ-	reg-	transform-
intens-	one-	rel-	transvers-
interact-	oper-	resist-	two
level-	or	sample-	typ-
kinet-	parameter-	satur-	under
iinear-	part	second-	used
longitudin-	plan-	shear-	using
loss-	point-	shift-	valu-
low-	pot-	sign-	var-
measur-	power-	siz-	vari-
measurem-ents	process-	stud-	veloc-
mechan-	propag-	suscept-	vibr-
method-	propert-	techniqu-	X
microscop-	pur-	term-	
model-	rang-	test-	
moment-	rat-	The-	

either, since the following pairs of stems "Al" and "alumin-", "Co-" and "cobalt-", "Ni" and "nickel-", or even "*deg*C" and "*deg*K" appear in the same group (i.e., with similar RR_s values).

Note that RR_s is not a reflection of stem frequency in the total inverted file but rather is only a measure of stem frequency within the document indexes containing the stem. RR_s of "maser-" is higher than that of "laser-". However the frequency of the stem "maser-" is 1154 (referring to 215 distinct documents) and that of "laser-" is 5197 (referring to 1252 distinct documents). Probably another measure, for example "RR_s times the stem frequency", may indicate actual degree of influence caused by the redundancy. Since we are not necessarily considering the redundancy as a harmful problem (see Section 2.7), only a list of stems by RR_s ranges was shown.

Table 3.6 itself may be of some help in organizing a thesaurus.

3.4.3 Redundancy Reduction in High Frequency Stems

Redundancy of these high frequency stems decreased as seen in Fig. 2.12. The amount of reduction, however, differs significantly for each stem. It depends on the type of the stem as mentioned in the previous section. Figure 3.1 shows examples of redundancy reduction for various stems. Highly redundant stems (like "film-") have greatly reduced their redundancy but,on the other hand, other high frequency stems (like "magnet-") have not necessarily reduced redundancy as much. Common stems (like "an") have remained fairly constant in their redundancy value.

3.5 High Frequency Common Stems

A list of high frequency common stems can be compiled by investigating low RR_s stems or by picking up stems with only one (or even no) ending. They are: an, between, due, near- (near, nearly), or, The- (The, They), used, using. Those stems may be culled out from the inverted file without loosing retrieval efficiency. Total frequency of those 8 stems is 17,734, 1.5 percent of total stem occurrences. Table 3.7 is a suggested list of expanded common words to be culled out.

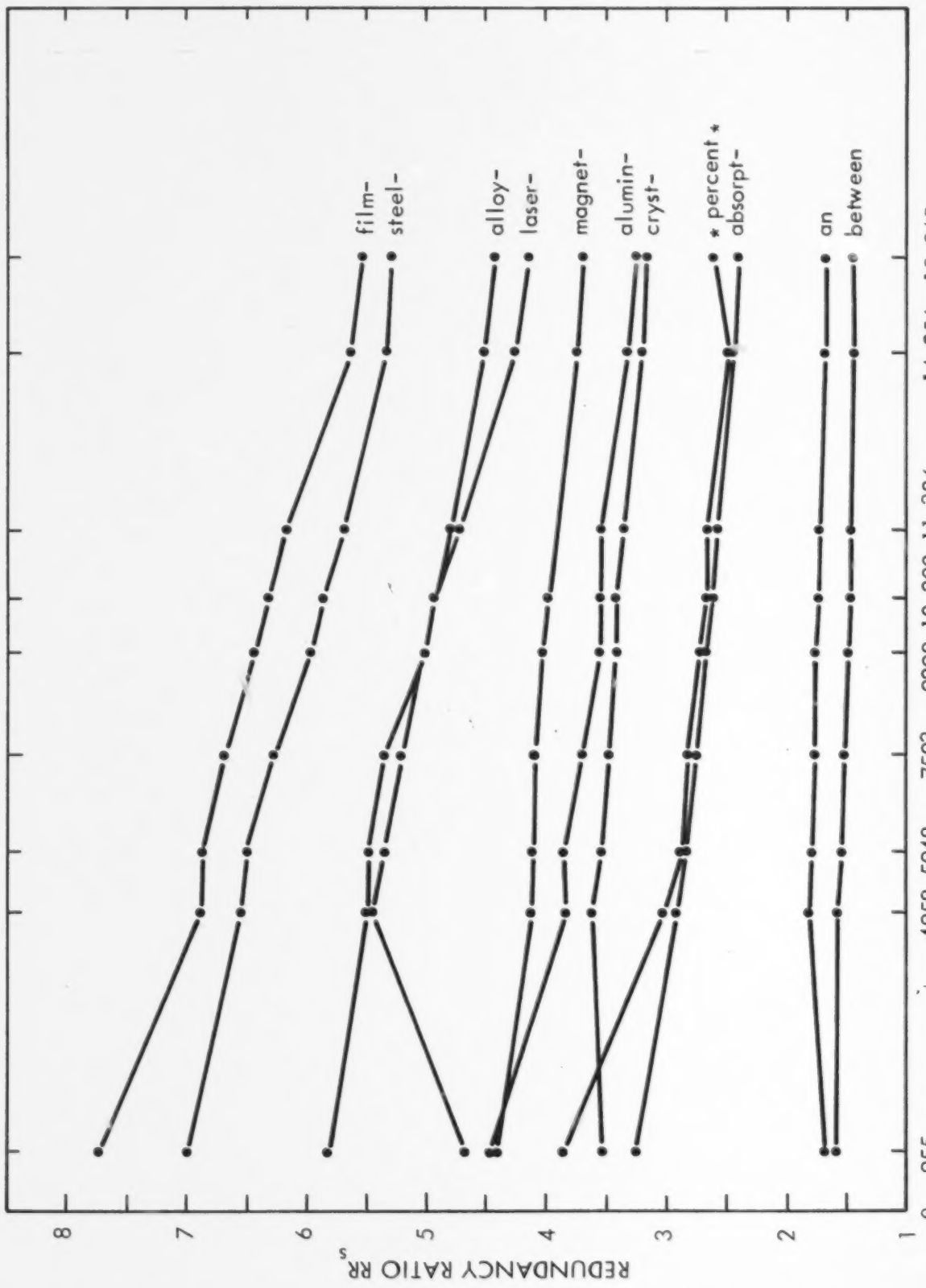

Fig. 3.1 Redundancy Ratio RR$_s$ for High Frequency Stems

Stems "one- (one, ones)" or "singl-e" are common too. But they are sometimes used as a qualifier, as in "single crystal", in which "single" has technical meaning. Culling this type of common stem from the inverted file would definitely cause loss of necessary information.

Table 3.7
Suggested Expanded List of Common Words
to be Culled out of Inverted File

a*	an	and*
as*	at*	between
by*	due	for*
from*	in*	near
nearly	of*	on*
or	the*	The
They	to*	used
using	with*	

* Indicates common words currently being culled out.

4. STEMS WITH EXTREMELY LOW FREQUENCY

4.1 Introduction

This chapter discusses extremely low frequency stems, especially those stems with only one occurrence. Table 2.1 indicated 75 percent of all stem types have occurred less than 5 times each. Figure 2.3 showed that 13,085 stem types, or 48 percent of all stems, have been used only once. This chapter delineates their characteristics.

4.2 Generation of Extremely Low Frequency Stems

4.2.1 File Updating and New Stems

When the Intrex collection was extended from 13,776 to 14,256 documents, updating statistics were gathered in detail. Table 4.1 shows the update changes. Comments associated with Table 4.1 are:

(1) The updated data base statistics should have incorporated 14,258 documents not 14,256. Actually, there were 14,258 documents in the catalog record file after the update. This sometimes happens because of small procedural errors or corrections to errors that cause count discrepancies.

(2) There were 3,254 stem types that merged into 25,496 stem types. Since 379 stem types out of 3,254 were entirely new, the updated inverted file included 25,845 stem types.

(3) By the same merging process, 455 word types out of 4,299 were entirely new.

(4) One occurrence of an unknown stem was observed since 116,934 = 1,090,051 + 26,822 + 1. This one may be the null stem mentioned in Section 2.4.

Table 4.1

Inverted File Updating from 13,776 Document Collection to 14,256 Document Collection

	Former Data Base	Added Document Collection	Updated Data Base
Number of Documents	13,776	482	14,256 (1)
Number of Stem Types	25,496	3,254	25,875 (2)
Number of Word Types	33,426	4,299	33,881 (3)
Total Number of Stem Occurrences	1,090,051	26,822	1,116,934 (4)
(1) - (4); see comments in the text.			

While 482 documents were added, 379 entirely new stem types appeared. It accounts for 0.8 new stems per document (see also Section 2.2.1). By manually comparing inverted file listings it was found that at least 309 stem types out of 379 new stems have a frequency of 1. This means each of those 309 stem types has been used only once in the indexing of the 482 documents. The average frequency of the other 70 stems is 2. The next section will discuss those new and isolated 309 stems.

4.2.2 Classification of New Stems

Entirely new and isolated stems were classified so that we could get information on "how" they were caused. Technical words, chemical formulae, and various symbols amount to 55 percent. Proper nouns account for 26 percent, a percentage higher than anticipated. Misspelled or meaningless stems amount to only 15 percent, much less than anticipated. (See Table 4.2)

Table 4.2

Classification of New and Isolated Stems
(309 newly appeared stems)

	Number of Stems	Subtotal	
Chemical Formulae	65 (21%)	98 (32%)	171 (55%)
Chemical or Material Names	6 (2%)		
Other Technical Words	27 (9%)		
Symbols	46 (15%)	73 (24%)	
Numerics	27 (9%)		
Proper Nouns	79	79 (26%)	79 (26%)
Household Words	12	12 (4%)	12 (4%)
Misspellings	33	33 (11%)	47 (15%)
Words with Delimiter Problems	14	14 (5%)	

The following is a grouped listing of these new and isolated 309 stems with comments for each particular group.

(a) Chemical Formulae (65)

eta(MgZn*sub 2*,
*sup 5*E*sub g*(*sup 5*D,
*sup 5*T*sub 2g*(*sup 5*D,
AgF*sub 2*, AlA-s, C*sub 48*K, C*sub 48*Rb,
C*sub 4h**sup 6*, CaF*sub 2*:Ag*sup 2+*,
CF*sub 3*)*sub 2*NO, CF*sub 3*)*sub 2*NOH,
Co*sub 2*T-i, CoO*sub 1 - x*, Cr*sup 5|*,
ellipsometry:SiO*sub 2*,

Fe*sub 2*(MoO*sub 4*)*sub 3*, He*star*,
He*sup 0*, Hf)*sub 2*, Hf*sup 181*, HFHOH,
HOOH, IrF*sub 6*, KBr:CN, KCl:*sup 6*L-i,
KCl:*sup 7*L-i, KCl:OH, KI:CN, KTaO,
Li*sub x*Me*sub 1 - x*O, LiCo*sub 1 - x*O,
LiNbO, LiNbO*sub 2*, M*sub 2*C, M*sup +*,
MgNi*sub 2*, MoF*sub 6*, NiO*sub 1 + x*,
NiO*sub 1 - x*, OsF*sub 6*, p*sup +*,
Pb(NbO*sub 3*)*sub 2*, Pb*sub 2*B-i, PtF*sub 6*,
R*sup +*, ReF*sub 6*, RhF*sub 6*, Rn,
RuF*sub 6*, SH*sup -*, SrF*sub 2*:Ag*sup 2+*,
Ta*sub 2*O*sub 3*, Ta*sub 2*O*sub 6*,
Th*sub 2*Ni*sub 17*, Th*sub 7*Ni*sub 3*,
ThNi*sub 2*, ThNi*sub 5*, Ti, Zr)C,
TiTaO*sub 4*, UF*sub 6*, V*sub Ni*Ni*sup 3+*,
V*sup 48*, VTaO*sub 4*, WRe,
YaFe*sub 3*O*sub 12*

Chemical formulae were expected to account for the majority of new stems, but they actually are only 24 percent of the total. They may be of little use for retrieval purposes (Who tries "subject Li*sub x*Me*sub 1 - x*O"?) but it is also true that each of them carries a certain amount of important (in-depth) information.

Some of the formulae listed look strange because of their unbalanced parentheses which was caused by a simplified word shredding routine. If a user issues a command "subject (HF)*sub 2*", it matches the stem "HF)*sub 2*".

Some of the items grouped here for convenience have only a structural resemblance to chemical formulae as, for example, "*sup 5*E*sub g*(*sup 5*D", but they are not chemical formulae.

(b) Technical

(b1) Chemical or Material (6)

bamboo, bis(hexyloxy, electronickel, methylbutoxy, orthoniob-ate, polyanhydroglucos-e

(b2) Other Technical Terms (27)

anharmonic-ities, cohesion-less, colinear,
counterelectrod-es, degress-ion,
electroneutr-ality, fluctuon-ic
flyback, homojunct-ion, interlink-age,
krytron, macroparticl-es, macroplast-ic,
nonideal-ized, nonreflect-ive, optometer,
preaver-aging, preconvect-ive, preillumin-ation,
pseudostat-istical, quadrantal, radioheliograph,
reticl-e, semiax-es, siegnetomagnon-s,
unidomain, unitaris-ed

The distinction between subgroups (b-1) and (b-2) is that the former contains chemical or material oriented stems.

A unique feature of this group is that most of the stems are of a compound form created by combining two or more components such as "electro+nickel", "electro+neutr-ality", etc. The Intrex stemming procedure works on a suffix, but not on these compound words nor on a prefix.

It is interesting to design a procedure that cuts off the prefix so that "pseudostat-istical" can also be "pseudo-stat-istical". Then the user will be able to get "pseudostatistical" too by simply asking "subject statistical". For "subject plastic", the same procedure will retrieve not only "plast-ic" but also "macro-plast-ic". If a user issues "subject reflective", the current Intrex system will retrieve documents about "reflect-ive". With a prefix procedure, it will also retrieve "non-reflect-ive" if that had been keyed as the unhyphenated string "nonreflective". Such a procedure may cause broader results with many irrelevant documents. But note that the Intrex system now does not check whether the context is "not reflect-ive" or "is reflect-ive" or is "non-reflective."

(c) Symbols (46)

1015A, 102A, 118A, 118L, 14R, 1673A,
1702B, 18B, 250B, 260A, 2N200, 3A8, 3T,
403A, 608F, 71A, 75C, 845AB, 8708A,
9558AQ, BX4, CTR, DH242, DIKS, EAI,

EBZ, FRA, G51, HP614A, HP616A, HP618A,
HPA, IZV, KN, KN22, L119, MBI, MIN,
N155, OSX, R201, S20, SD6, UHW6, VM, XDR6

The lexical structure of these symbols is very simple. They consist of a sequence of digits and a sequence of capital letters. One exception "MIN" (a type of a microscope), may be misspelled from "MLN".

These stems have special meanings in Intrex indexing. Most of them were associated with range 4 terms. Range 4 terms express a laboratory tool or theoretical technique referred to in the document and not otherwise named within any other index term. Actually all of the symbols here concern tools (but not theoretical techniques which are more likely to have names rather than symbols) such as "General radio types 1673A and 1620A bridges (4);" or "Bolex model BX4 moving picture camera (4);". Three exceptions were "2N2000" (type of diode) which was associated with a range 2 term, "Type 2N2000 diode-connected transitions (2);". "DH242" and "N155" were used within a range 1 term though they are commercial names, "Kinetics of selective oxidation of chromium in commercial Ni-(Fe)-Cr superalloys, N155, DH242, Hastelloy X, and TDNiCr (1);".

(d) Numerics (27)

0.00063, 0.1715, 0.46, 0.04, 1.10, 1218,
13.5, 144, 16.5, 1760, 18.0, 2.15, 2.41
2.9, 2475, 2829, 3030, 341, 3470, 39.9,
465, 485, 6.6, 7242, 77.2, 78.8, 9.15

These numerics may be divided into two groups, numbers with decimal points and numbers without decimal points (integer). Those with decimal points are used with various units (*percent*, eV, MHz, GHz, *deg*K, etc.). There is no special range in which they appear more frequently. Most of those without decimal points are parts of symbols ("1218" from GR-1218-A oscillator, "2475" from KODAK 2475 film, etc.). Consequently they are used in range 4 terms. But some of the integers function the same as the decimal points numbers as in: 1760 *deg*C and 3470 *A*.

Another characteristic is that most of these numerics contain at least three digits each. It means when a two digit number appears in indexing terms, probably it had already been used elsewhere (in indices for other documents). However it is not correct to assume that each occurrence of a two digit number represents the same context. A randomly selected numeric, "5.5", has been investigated. It contains only two digits and it was used in nine different documents (indices).

5.5 pct Ni
bands 1-1.5 GHz and 3.5-5.5 GHz
energy range 0.6 - 5.5 eV
4.5 *deg* and 5.5 *deg*K
the low energy 5.5- eV exciton
between 1.8 and 5.5 eV
1.4 *percent* and 5.5 *percent*
a 5.5 mole *percent* solution
5.5 power telescope

"5.5" cannot be regarded as having a constant context. It is a magnitude whose relative meaning depends upon a unit of measurement which may usually be found nearby in the above phrases, as for example, eV, GHz.

(e) Proper Nouns (79)

Ame-s, Auzel-'s, Bab-o, Balmer, Batschinsk-i,
Bayne-'s, Benney-'s, Bolex, Breslar, Chandr-a,
Chrenk-o, Cittert, Cleary-'s, Costain, Cromel,
Crowell, Delbecq, Doolittle, Drabkin-'s,
Dyzaloshinsk-i, Edmond-s, Ektachrom-e,
Elmore-'s, Falhenhag-en, Frieder, Gigerflex,
Grashoff, Hoch-'s, Hornstra-'s, Hou, Kollmorgen,
Kozak, Lynd-en, Lynn, Mahler-'s, Mattossi-'s,
Mikoshiba-'s, Mikusinski-'s, Monroe-'s, Nelder,
Niemyski-'s, Nummendal, Oosting-'s, Oshman,
Padovani, Pascal, Pathria, Pitayevsk-i, Prakash,
Primak-'s, Pyun, Quad, Redei, Ribeir-o, Rideout,
Rie-s, Rocketdyn-e, Ryum-'s, Satterthwait-e,

Saxena-'s, Schroer, Sekera-'s, Slutsk-y,
Spectraphys-ics, Stakelbeck, Stathem,
Stratton, Suzuoka-'s, Syoz-i, Tabuchi-'s,
Taupin, Taxman, Trappen-iers, Varitemp,
Vibrox, Warburg, Wehner, Widmanstall-en,
Workman

Proper nouns were not expected to appear with such frequency in newly created stems. Among them, personal names account for more than 80 percent. Trade marks ("Ektachrom-e"), company names ("Kollmorgen"), and model names ("Vibrox") comprise the remainder.

Proper nouns mainly appear in index terms ranged 2, 3, or 4, and sometimes 1. Personal names sometimes appear in range 1 in the form of a named classic or widely known theory or method. On the other hand, though range 4 is for theoretical techniques referred to in the document, personal names do not necessarily appear there with any great frequency. This is mainly because those techniques are usually mentioned within a broader context in terms with ranges 2 or 3.

(f) "Household" words

addend-a, delineat-ed, enhancem-ents,
entrust-ed, fingerprint, neighborhood-s,
put, semant-ic, slugg-ish, softest,
spiral-ing, suppos-ing

It was surprising to find that these general words, which are correctly spelled, appeared only once in the file. Some of these words, while grouped here as general words, could also have special meaning as technical terms.

Half of these isolated stems were created by the Intrex stemming procedure. For example, though the stem "enhancement-" has often been used, the word "enhancements" created a new stem "enhancem-" with ending "ents". Stems "deline-" from "delineate", "neighbor-" from "neighborhood", "slug-" from "slug", "soft-" from "soft", and "spir-" from "spiral" could not incorporate words "delineated", "neighborhoods", "sluggish", "softest", nor "spiraling".

"addenda" is rarely used. "fingerprint" or "semantic" are suprise appearances considering the specific scope of the current data base. As for the remainder "put", "entrusted" and "supposing", all the author can say is that this kind of new household word will continuously appear in our inverted file.

(g) Stems of no use

(g-1) Misspellings

(Correct spelling is shown in parenthesis.)

Arhhenius (Arhenius), attena (antenna), Baush (Bausch), Boltzmannn (Boltzmann), chromiun (chromium), cryt-als (crystal), Davydow (Davydov), deporalis-ed (deporalized), deglcin-e (deglycine), electrodesposit-ed (electrodeposit-ed), enviroment (environment), express-ions (expressions), Feyman (Feynman), geometric-es (geometrics), Heinsenberg (Heisenberg), incubatron (incubation), infinetesimal (infinitesimal), intergranl-ar (intergranular), intiner-ant (itinerant), intris-ic (intrinsic), kryt-on (krypton), magnetoelect-ic (magnetoelectric), oscilll-ation (oscillation), phass-e (phase), phenonen-a (phenomen-a), photomulipl-ier (photomultiplier), polyrpopyl-ene (polypropyl-ene), regor-ous (rigorous), resistv-ity (resistivity), resonn-ance (resonance), sensitivet-y (sensitivity), titant-e (titanate), ultrasoudn (ultrasound)

The author tried to detect misspellings within the scope of information in the INTREX file. Some of them ("Boltzmannn", "enviroment", etc.) could easily be recognized. Others, "Heinsenberg" for example, were judged misspelled because the context said "... isotropic Heinsenberg ferromagnet and antiferromagnet", but a search on "subject Heisenberg ferromagnet antiferromagnet" found 7 documents. "intergranl-ar" was judged misspelled because a word "intergranular" was found in the same subject index, and a search "subject intergranular" retrieved 44 documents.

But no attempt was made to trace words outside of the file. "MlN" of group (c) may have been "MLN" if it had been correctly spelled, since the context is "MBS-1 and MlN-6 microscope". "MlN" or "MLN" was used only there, so that within the scope of information of the inverted file, and the catalog file the author could not decide whether it was misspelled or not.

Most of these misspellings are long strings (from 5 characters to 17). There is no misspelling of the first character. These 33 misspellings out of 3,254 stems amount to one percent.

(g-2) Stems with delimiter problems

*deg*0, *deg*1, *degree*, *degree*K,
3K/5U, 95Mc/sec, g/cm*sup 3*,
I4*sub 1*/-a, kgf/cm*sup 2*,
sub 1 - x)*sub 5*O*sub 15*,
COOH)*sub 3* CaCl*sub 2*, KH*sub 2*PO*sub 4*b-y,
approx-i, C*sub 2[*nu*]

These stems involve various problems. First, stems that contain "/" cannot be used as search terms, since "/" is a delimiter in the Intrex command system. "/" is now being used for "÷" in special character coding. This should be amended.

Second, misspacing in special character coding causes problems. For example, "sub 1 - x*)*sub 5*O*sub 15*" is part of a longer and meaningful chemical formula $K_3Li_2(Ta_xNb_{1-x})_5O_{15}$. A space has been put by mistake between "Nb" and "*sub 1 - x*" and it caused the partial meaningless stem. The document would not be retrieved when a user issues a correctly spelled subject search. "KH*sub 2*PO*sub 4*b-y" is another example. In this case, a space should have been put between "*sub 4*" and "by". The document cannot be retrieved by correct subject search either. "*deg*0" and "*deg*1" are a part of "00 *deg*1-10 *deg*0 band" which looks like misspacing too. The hyphen is a delimiter in the file generating program.

"COOH)*sub 3*CaCl*sub 2*" was generated from the chemical formula $(CH_3\text{-}NH\text{-}CH_3\text{-}COOH)_3CaCl_2$ by the same hyphen problem. However since the hyphen is also a delimiter in command program, a search command subject (CH*sub 3*-NH-CH*sub 3*-COOH)*sub 3*CaCl*sub 2* successfully retrieves the document in this case.

"approx-i" was found in the context "Hartree approxi- mation". Input miskeying must have caused it.

The document itself is accessible by "subject approximation", since the word "approximation" appears in the same index term set.

There are three stems caused by miscoding (or misspelling) of special characters. Both "*degree*" and "*degree*K" should be "*deg*" or "*deg*K", and "C*sub 2[*nu*] " should be "C*sub 2[*nu*] *".

4.3 Extremely Low Frequency Stems in the Whole Inverted File

The preceding section classified new and isolated stems. The next step was to investigate the full inverted file, that is, to investigate both old and new isolated stems. The inverted file for the 14,256 document collection included 12,323 isolated (frequency one) stems and 321 stems were selected from them by the following procedure:

(1) The 450 page listing of those isolated stems was available.

(2) 12 pages were extracted from it by selecting every 38th page.

(3) All stems from those 12 pages were selected.

Table 4.3 shows the result of classification. Instead of listing all stems, a discussion alone follows, comparing these results for the whole file with those of Table 4.2 for an incremental collection.

The general trend of Table 4.3 is similar to that of Table 4.2. The percentage of chemical formulae and technical words increased from 32% to 40% while symbols and numerics decreased from 24% to 15% in the whole file sample. These changes may have been caused by the sampling procedure and this is discussed in a later paragraph. The remaining categories underwent little or no change in their percentages. Proper nouns account for one quarter of the total isolated stems. Useless stems (misspellings and stems with delimiter problems) account for 16% of the isolated stems. Household-type words still reflect only 4% of the total, although again a substantial number in this set really reflect stemming algorithm inadequacies as in "fastest" not stemming to "fast-" and "Laboratory-'s" not stemming into "labor-".

Table 4.3

Classification of 321 Isolated Stems
(from the whole inverted file)

	Number of Stems	Subtotal	
Chemical Formulae	80 (25%)	130 (40%)	177 (55%)
Chemical or Material Names	12 (4%)		
Other Technical Words	38 (12%)		
Symbols	28 (9%)	47 (15%)	
Numerics	19 (6%)		
Proper Nouns	81	81 (25%)	81 (25%)
Household Words	12	12 (4%)	12 (4%)
Misspellings	33	33 (10%)	51 (16%)
Words with Delimiter Problems	18	18 (6%)	

The number of stems grouped under chemical formulae but which really represent other special characters increased to 20% of that category (16 stems out of 80) from 3% (2 out of 65). These are, for example, "*alpha**sub m*", "*chi**sub ee*(q", and "*chi**sub *parallel**". The number of technical words increased to 16% from 11%. These increases were at the expense of a decreased percentage of symbols and numerics but both changes may have been the result of a sampling bias in selecting isolated stems from the inverted file. The file listing is based upon stem arrangement by computer collating sequence. Those stems beginning with the character "*" appear in a cluster and those which begin with numerics

appear in another cluster. The systematic sampling by selecting whole pages causes a bias because of these clusters. However, the main point of discovering the kinds of isolated stems is not obscured by this bias.

Another good example for a prefix procedure appeared in the sample. There are 17 isolated stems which begin with the prefix "multi-" out of more than 70 such stems in the file beginning with that prefix. Some of them are "multicenter", "multimagnon", "multidomain", and "multimode".

The results of this classification over the whole inverted file follows in the same direction as that discussed in Section 4.2.2. It is different from what was anticipated. Most misspelled words had been manually cleared up before updating the inverted file. We may be able to clean up the residue missed by oversight but that only reduces by 15 percent the total number of isolated stems or less than 1 percent of the whole inverted file storage space.

If a user makes a subject search using one of these extremely low frequency stems (words), he will get only one document. In one sense, this is because his subject is pretty specific. The matched stem can be said to be specific too. There is high possibility the user judges the document relevant. Proper nouns are good examples of this specificity. They caused a number of new stems and they represent specific aspects of a document which may not be indexed under controlled vocabulary.

It is interesting to note that the National Library of Medicine's vocabulary control list "Medical Subject Headings (MeSH)" is continuing its growth and that it incorporates a special group of proper nouns, "geographic headings". The 1964 version of MeSH (3rd Ed.) contained 5, 812 subject headings. In 1967 the number grew to 6, 800.[11, 14] Though precise data after that period is not available, the 1970 version of MeSH reports that 162 new headings are included (and 6 deleted) that year.

A special set of place names "Geographical Headings" exists beside MeSH main headings and subheadings. There were 149 geographical headings in 1964, 264 in 1967. These headings are not listed

in the formal annual publication of MeSH, but only in special versions of it ("MeSH Tree Structures" or "MeSH Alphabetic List"). The reason why MeSH only includes place names (excluding personal names) is not clear. Possibly, frequently used proper nouns in the medical field are listed in MeSH itself. However the point here is that proper nouns, even if limited only to geographic place names, have been incorporated into MeSH.

If 13,085 isolated stems in the May 1971 version of the Intrex inverted file were eliminated, the resulting file would contain 14,206 stem types. If all stems of frequency not more than 5 were eliminated, the resulting file would include only 6,778 stem types. This number is close to the number of subject headings in MeSH, although those headings are often composed of more than a single word, thereby increasing the actual number of word types in their index.

5. SUMMARY

Characteristics of the INTREX subject/title inverted file were studied and various aspects of the INTREX indexing method were clarified. The latest available version of the file from May 1971 covering 15,845 documents was given primary attention but earlier versions were also examined to study some characterstics as a function of file growth.

A document is assigned on the average, 10 index terms (9 if the title is discounted), with 7.6 words per term (excluding common words on a stop list). These 76 words in the complete inverted file index set (including title words) contain 39.1 unique stem types. The average length of a stem type is 7.8 characters.

There are 27,291 stem types and 35,645 word types in the inverted file. A stem type thus represents an average of 1.3 word types. However, 84% of the stem types represent only 1 word type, 10% represent only 2 word types, and 3% represent just 3 word types. Any study of stemming should concentrate on the retrieval effectiveness of the 16% of the stem types which represent 2 or more word types.

The average number of occurrences (tokens) of a stem type, that is, its frequency, is 44.1. However 48% of the total number of stem types have a frequency of 1, and 75% have a frequency of less than 5. At the other end of the scale, 260 stems (0.95%) occur with a frequency greater than 1000 and the occurrences of these stems account for 54% of the 1,204,282 total stem occurrences. The most frequently occurring stem is "magnet-" with 16,824 occurrences. Because of the redundant nature of natural language indexing, the number of occurrences per stem type is not the same as the number of documents referenced per stem type, the average of which is 22.6 documents.

The distribution of stem occurrences by the range number assigned to the index terms in which they occur was investigated. We find that 2.8% of the stems have occurred in terms assigned range

number 0, 14.7%, 33.1%, 28.9%, and 11.1% are in terms assigned ranges 1, 2, 3, 4 respectively, and 9.3% are title occurrences assigned range 5.

A Redundancy Ratio (index words per document/stem types per document) was calculated to be 1.95 for the total inverted file. This is a cumulative value just as are other values in this summary. Values representative of documents indexed in any given time period may differ significantly from those of another period. A single cumulative value does not indicate the long-time trend. Changes in several of the parameters under study were examined during different stages of file growth. For what is essentially the just 5500 documents indexed, the average document had a total index of 101.5 words with 45.4 stem types, yielding a redundancy ratio of 2.23. For what is essentially the latest 10,000 documents included in the inverted file, there were 62.7 index words and 35.8 stem types yielding a redundancy ratio of 1.75. Most of this decrease was brought about by an intentional change in the indexing process to reduce redundancy. An uninvited byproduct of the effective reduction in redundancy was a 21% decrease in the average number of stem types per document which represents a potential loss of information.

During a time period (1968 to 1971) when the document collection grew by a factor of 17 (from 955 to 15,845), the number of stem types increased only by a factor of 4 (from 6,700 to 27,291). In that same period the number of new stem types added to the inverted file during updates decreased from 6.7 to 0.85. The current rate of 0.85 new stem types appearing per document added to the data base is now at a leveling point or at least very close to one. Nevertheless, the number itself is somewhat higher than expected and it has been shown that proper nouns, formulae, and compound technical terms appearing in the free vocabulary are their main source.

Particular attention in the analysis was given to those stems with extremely high frequency and with extremely low frequency. The 260 high frequency stems (those with more than 1,000 occurrences each) are categorized into three groups: (1) purely technical words; (2) general words sometimes used technically; (3) common, non-

technical words. A useful distinction between groups (1) and (2) is that the technical words could be used as a single search word, whereas a word from the more general group, such as "field-" or "effect-" could not be meaningfully used by itself. A ranking of high frequency stems at several stages of inverted file growth shows little or no change in the rank ordering. Unless drastic changes occur in data base subject matter, stems having high frequency in the early stages of file growth are most likely to remain high in frequency and in the same relative order. The three stems ranking highest in number of occurrences in the total file are "magnet-", "temperatur-", and "effect-". High frequency stems were also considered in terms of their redundancy on the basis of document references. Another redundancy ratio was defined for each stem as (frequency/number of distinct document references). A ranking of stems by decreasing redundancy ratio shows that "film-", "maser-", and "steel-" are on top with a redundancy ratio of between 5.0 and 5.9; these three stems ranked 21, 220, and 60, respectively when considered only in terms of their total number of file occurrences. The most frequent stem, "magnet-" has a redundancy ratio of between 3.0 and 3.9 while "temperatur-" and "effect-" have redundancy ratios between 2.0 and 2.9. Synonyms such as "alumin-" and "Al" or "therm-" and "heat", were found to have very similar redundancy ratio values, whereas these synonyms do not necessarily appear near each other in a rank order by occurrence only. Less-technical words have a lower stem redundancy ratio than do more-technical words. It is hypothesized that an inverse correlation may exist between these stem redundancy ratios and the generality of word usage in the indexes to the collection. The stem redundancy ratios for technical high frequency stems have been shown to be steadily decreasing (as much as 30% in some cases) as the file has grown. The stem redundancy ratio for common word high frequency stems has decreased only slightly in the same time period, suggesting further their more distributive nature in the index term sets.

For optimal retrieval effectiveness, it is intuitive that search terminology should begin with the most precise vocabulary that still

yields high recall. Frequency counts and, possibly, frequency and redundancy ratios too, are a useful aid to a searcher organizing his strategy. Counts are already included within the inverted files and as feedback to a user; ratios could be a similar candidate if their value to optimizing a search strategy can be established.

Stems with extremely low frequency were also analyzed in some detail. Because there are 13,085 stem types that have occurred only once, the inverted file was sampled, and the characteristics of this sample were also compared with those of all 309 new single occurrence stems in a recent update batch. Those stems were categorized. The percentage totals for each category from the update batch agree reasonably well with that for the total file. The data from the update batch is summarized here. Some 32% of these new stems were derived from technical words (21% were chemical formulae, 2% were chemical or material names, and 9% were other technical words), 24% were either symbols or numerics, 26% were proper nouns, 11% were misspellings, 4% were general words, and 5% were caused by delimiter problems. Personal names account for more than 80% of the proper noun category. Many of the new technical name or word stems are of compound words formed with prefixes. This suggests that a prefix stemming procedure may have potential benefits.

How well does the inverted file, whose characteristics were reported here, represent the search expressions of users? One step toward answering that question is to analyze those search expressions along similar lines. These search expressions are currently recorded in monitor files of user search sessions. However, a program could be designed to record, cumulate, and calculate search expression characteristics for given time periods, perhaps a month. Statistical reports prepared for each period could be cumulated. In this way, we could then know what types of items in the inverted file are actually used for computer search purposes.

REFERENCES

1. Reintjes, J.F., "System Characteristics of Intrex", 1969 Spring Joint Computer Conference, Boston, Mass., May 14-16, 1969, AFIPS Conference Proceedings, Vol. 34, pp. 457-459.

2. Benenfeld, A.R., "Generation and Encoding of the Project Intrex Augmented Catalog Data Base", Proceedings of the 1968 Clinic on Library Applications of Data Processing, Urbana, Illinois, University of Illinois, Graduate School of Library Science, 1969, pp. 155-198.

3. Marcus, R.S., P. Kugel, and R.L. Kusik, "An Experimental Computer-Stored, Augmented Catalog of Professional Literature", 1969 Spring Joint Computer Conference, Boston, Mass., May 14-16, 1969, AFIPS Conference Proceedings, Vol. 34, pp. 461-472.

4. Lovins, J.B., "Error Evaluation for Stemming Algorithms as Clustering Algorithms", Journal of the American Society for Information Science, Vol. 22, No. 1, January - February 1971, pp. 28-40.

5. Lovins, J.B., "Development of a Stemming Algorithm", Mechanical Translation, Vol. 11, 1968, pp. 22-31.

6. Marcus, R.S., A.R. Benenfeld, and P. Kugel, "The User Interface for the Intrex Retrieval System", Presented at the Workshop on the User Interface for Interactive Search of Bibliographic Data Bases, Palo Alto, California, January 14-15, 1971. (Proceedings to be published by AFIPS Press)

7. Kusik, R.L., "A File Organization for the Intrex Information-Retrieval System on the 360/67 CP/CMS Time-Sharing System", ESL-TM-415, Electronic Systems Laboratory, Massachusetts Institute of Technology, Cambridge, Mass., January 1970.

8. Project Intrex, Semiannual Activity Report, PR-9, Massachusetts Institute of Technology, Cambridge, Mass., 15 March 1970.

9. Uemura, S., "New Calculations of Subject-Word Statistics", Electronic Systems Laboratory Memorandum INTREX-70, December 18, 1970. (Unpublished)

10. Project Intrex, Semiannual Activity Report, PR-6, Massachusetts Institute of Technology, Cambridge, Mass., 15 September 1968.

11. Austin, C.J., "MEDLARS 1963-1967", National Library of Medicine, Bethesda, Maryland, 1968.

12. Lovins, J.B., "Error Evaluation for Stemming Algorithms as Clustering Algorithms", ESL-R-411, Electronic Systems Laboratory, Massachusetts Institute of Technology, Cambridge, Mass., December 1969.

13. Project Intrex, Semiannual Activity Report, PR-12, Massachusetts Institute of Technology, Cambridge, Mass., 15 September 1971.

14. "The Principles of MEDLARS" National Library of Medicine, Bethesda, Maryland, 1970.

Appendix A

LSTDST Program Listing

```
BEGIN
COMMENT  ************************************************************************
  PROGRAM-NAME  LSTDST
  COMMAND       R  LSTDST  SEG-NO1  FILE-NAME  SEG-NO2  DOC-NO  -INTVAL-
  FUNCTION      THIS PROGRAM PRODUCES A TABLE OF STATISTICS OF THE INVERTED FILE
                AND A LIST OF STEM FREQUENCY DISTRIBUTION.
  NOTE  1.      TWO SEG-NO'S DESIGNATE THE FIRST AND LAST SEGMENTS TO BE PROCESSED.
                SEG-NO1 IS USUALLY 1.  SEG-NO2 CAN BE ANY INTEGER.  IF IT EXCEEDS
                THE ACTUAL NUMBER OF SEGMENTS, THE WHOLE INVERTED FILE IS PROCESSED.
                FILE-NAME SHOULD BE THE LATEST SECONDARY NAME OF THE INVERTED FILE.
                DOC-NO SHOULD BE THE CURRENT NUMBER OF DOCUMENTS.  INTVAL  IS OPTIONAL.
                IF IT IS SPECIFIED, THIS PROGRAM PRODUCES ANOTHER LIST OF STEM FREQUENCY
                DISTRIBUTION WITH THAT FREQUENCY INTERVAL.  IF NOT SPECIFIED, THIS
                LIST WILL NOT BE PRODUCED.
        2.      THE OUTPUT NEEDS A LOT OF PAGES (MORE THAN 14 PAGES EVEN IF NO INTVAL
                IS SPECIFIED.)
        3.      STATISTICS FOR THE NUMBER OF ENDINGS DISTRIBUTION SHOULD BE COMPARED WITH
                THAT OBTAINED FROM IESTAT.  BOTH SHOULD BE IDENTICAL.  (TWO PROGRAMS
                TRACE THE INVERTED FILE IN DIFFERENT WAYS.)
************************************************************************************
     COMMENT 'LSTDST' IS A STATISTICS GATHERING PROGRAM INTENDED TO DETERMINE THE
             SUBJECT / TITLE INVERTED FILE LIST LENGTH FREQUENCY DISTRIBUTION. THE
             PROGRAM WAS WRITTEN BY MR. R. KUSIK, 1969, AND MODIFIED BY S. UEMURA,
             NOV. 1970, JAN. 1971. $,
               SYNONYMS
                    INTEGER = POINTER $,
               CONT .BCD. .CONT. $,
               SMON .BCD. .SMON. $,
               SMOFF .BCD. .SMOFF. $,
               INTEGER N1,N2,ENDN1,SEGNO,ENDSEG,MAXSIZ,EOFCNT,SECSIZ,SECSZC,OFFSET,
                LSTSIZ,IVLSIZ,IVMAX,NOLSTS,BIGLST,I,TAB,ERRCOD,T1,T2,T3,NOREFS,REFSK
                ,NODIES,DIESK,NOEDGS,NODOCS,AVRRD,AVRRL,AVRDD,INTVAL,IVIDC,SPABAC,
                NWORDS,EDSMAX,EDSDST,EDSSIZ,EDSITM $,
               POINTER WKBUF.,LIST.,DIST.,NAME.,IVDIST. $,
               BOOLEAN LAST $,
               PROCEDURE BEOPEN,BEREAD,BECLOS,TYPEIT,WELXA,DORMNT $,
               INTEGER PROCEDURE FREZ,LOCMES,.C.ASC,DEFBC,BCDEC,BCDASC,COMARG,BECODE
                ,FILCNT $,
               FILCNT $,
               ARRAY BUF1(432),BUF2(432),BUF3(432),WRKBUF(432) $,
               INTEGER COMPONENT RH18,ADCOM,EDS,DCL,NREF,C,CWL,BYN,DCOM,HEDR $,
               PACK 777777C,0,SPECIAL COMPONENTS RH18 $,
               PACK 77777C,0,ADDRESS COMPONENTS ADCOM,CWL,NREF,BYN $,
               PACK 77777C18,18,DECREMENT COMPONENTS DCOM,DCL $,
               PACK 37C27,27,SPECIAL COMPONENTS EDS $,
               PACK 7C33,33,SPECIAL COMPONENTS HEDR $,
               RH18 $=$ ADCOM $=$ C $=$ CWL $=$ DCOM $=$ HEDR $=$ 0 $,
               NREF $=$ DCL $=$ 1 $,
               BYN $=$ EDS $=$ 2 $,
               PRESET
                    BEGIN
                    MAXSIZ = 17000 $,
                    IVMAX = 1700 $,
                    EDSMAX = 20 $,
                    BIGLST = 0 $,
                    NOREFS = 0 $,
                    NOLSTS = 0 $,
                    NODIES = 0 $,
                    NOEDGS = 0 $,
                    DIESK = 0 $,
                    REFSK = 0 $,
                    SECSIZ = SECSZC = 432 $,
                    END $,
               LAST = FALSE $,
               SPABAC = (2 .LS. 18) .V. LOC 40010C18 $,
               TAB = .C.ASC(.C. /   /) $,
               N1 = COMARG(1) $,
               N2 = COMARG(2) $,
               ENDN1 = COMARG(3) $,
               NODOCS = COMARG(4) $,
               NODOCS = BCDEC(NODOCS) $,
               IF COMARG(5) EQL -377777777777C
               THEN INTVAL = 9999
               ELSE INTVAL = BCDEC(COMARG(5)) $,
               SEGNO = BCDEC(RH18(LOC N1)) $,
               ENDSEG = BCDEC(RH18(LOC ENDN1)) $,
               DIST. = FREZ(MAXSIZ+1) $,
               IVDIST. = FREZ(IVMAX+1) $,
               EDSDST = FREZ(EDSMAX+1) $,
               WKBUF. = LOC WRKBUF $,
     COMMENT $,
     COMMENT $,
ACTION $       BEOPEN(.BCD. /R/,N1,N2,BUF1,BUF2,BUF3,OPNERR) $,
NXTSEC $       BEREAD(N1,N2,WRKBUF TO SECSIZ,SECEND,EOFCNT,RDERR) $,
     COMMENT $,
```

```
MOVPTR $        IF (OFFSET = ADCOM(WKBUF.)) EQL 0
                THEN IF LAST
                     THEN GOTO NXTSEG
                     ELSE GOTO NXTSEC $,
                LIST. = WKBUF.+OFFSET $,
     COMMENT $,
NXTLST $        IF (LSTSIZ = NREF(LIST.)) GRT MAXSIZ
                THEN GOTO OVRFLO $,
                IF HEDR(LIST.) NEQ 7
                THEN TYPEIT(LOCMES(.C. /HEDR ERROR/),SEGNO,OFFSET) $,
     COMMENT ****** $,
                IVLSIZ = ((LSTSIZ-1)/INTVAL)+1 $,
                C(IVDIST.+IVLSIZ) = C(IVDIST.+IVLSIZ)+1 $,
     COMMENT ****** $,
                FDSSIZ = FDS(LIST.) $,
                NOFDGS = NOFDGS+FDSSIZ $,
                C(FDSDST+FDSSIZ) = C(FDSDST+FDSSIZ)+1 $,
                NODIFS = NODIFS+DCL(LIST.) $,
                NOREFS = NOREFS+LSTSIZ $,
                NOLSTS = NOLSTS+1 $,
                C(DIST.+LSTSIZ) = C(DIST.+LSTSIZ)+1 $,
     COMMENT $,
BACK $          LIST. = LIST.+CWI(LIST.) $,
                IF LIST. GEQ WKBUF.+SECSIZ
                THEN IF NOT LAST
                     THEN GOTO NXTSEC
                     ELSE GOTO NXTSEG
                ELSE GOTO NXTLST $,
SECEND $        IF EOFCNT EQL 0
                THEN GOTO NXTSEG $,
                LAST = TRUE $,
                SECSIZ = EOFCNT $,
                GOTO MOVPTR $,
     COMMENT $,
NXTSEG $        LAST = FALSE $,
                SECSIZ = SECSZC $,
                RECLOS(N1,N2,CLSERR) $,
                TYPEIT(SMON,SPARAC,CONT,SMOFF) $,
                SEGNO = SEGNO+1 $,
                IF SEGNO GRT ENDSEG
                THEN GOTO OUTPUT $,
                RH18(LOC N1) = DEFRC(SEGNO) $,
                NWORDS = FILCNT(N1,N2) $,
                IF NWORDS GRT 0
                THEN GOTO ACTION $,
                TYPEIT(LOCMES(.C. /*** NO MORE SEGMENT --/),SEGNO) $,
     COMMENT $,
     COMMENT $,
OUTPUT $        TYPEIT(LOCMES(.C. /   /)) $,
                TYPEIT(LOCMES(.C. /************************/)) $,
                AVRRD = 10*NOREFS/NODOCS $,
                AVRRL = 10*NOREFS/NOLSTS $,
                AVRDD = 10*NODIFS/NODOCS $,
REFRDC $        IF NOREFS GRT 100000
                THEN BEGIN
                     REFSK = REFSK+1 $,
                     NOREFS = NOREFS-100000 $,
                     GOTO REFRDC $,
                     END $,
DIFRDC $        IF NODIFS GRT 100000
                THEN BEGIN
                     DIFSK = DIFSK+1 $,
                     NODIFS = NODIFS-100000 $,
                     GOTO DIFRDC $,
                     END $,
TYPALL $        TYPEIT(LOCMES(.C. /TOTAL NUMBER OF DOCUMENTS         /),NODOCS,LOCMES(
                 .C. / (YOU TYPED THIS VALUE, CHECK IT AGAIN.)/)) $,
                TYPEIT(LOCMES(.C. /TOTAL NUMBER OF LISTS (STEMS)     /),NOLSTS) $,
                TYPEIT(LOCMES(.C. /TOTAL NUMBER OF DIFFERENT WORDS   /),NOFDGS) $,
                TYPEIT(LOCMES(.C. /TOTAL NUMBER OF REFERENCES        /),REFSK,LOCMES(
                 .C. /* 100000 +/),NOREFS) $,
                TYPEIT(LOCMES(.C. /TOTAL NUMBER OF DISTINCT DOCS.    /),DIFSK,LOCMES(
                 .C. /* 100000 +/),NODIFS) $,
                TYPEIT(LOCMES(.C. /************************/)) $,
                TYPEIT(LOCMES(.C. /AVERAGE NUMBER OF REFS. PER DOCS./),AVRRD,LOCMES(
                 .C. ./ 10.)) $,
                TYPEIT(LOCMES(.C. /AVERAGE NUMBER OF REFS. PER LISTS/),AVRRL,LOCMES(
                 .C. ./ 10.)) $,
                TYPEIT(LOCMES(.C. /AVERAGE NUMBER OF D.D.  PER DOCS./),AVRDD,LOCMES(
                 .C. ./ 10.)) $,
                TYPEIT(LOCMES(.C. /************************/)) $,
TYPFDS $        TYPEIT(LOCMES(.C. /NO.FNDS.  NO.LISTS  /)) $,
                FOR I = 1 STEP 1 UNTIL FDSMAX
                DO   BEGIN
                     FDSITM = C(FDSDST+I) $,
                     IF FDSITM NEQ 0
                     THEN TYPEIT(TAB,I,TAB,TAB,FDSITM) $,
                     END $,
                TYPEIT(LOCMES(.C. /*****************************/)) $,
     COMMENT ****OUTPUT CHECK**** $,
```

```
                IF COMARG(6) NEQ -377777777777Q
                THEN BEGIN
                     IF COMARG(5) NEQ -377777777777Q
                     THEN GOTO OUTINT $,
                     END $,
                NOLSTS = 0 $,
                NOREFS = 0 $,
                REFSK = 0 $,
                FOR I = 1 STEP 1 UNTIL MAXSIZ
                DO   BEGIN
                     T1 = C(DIST.+I) $,
                     IF T1 NEQ 0
                     THEN BEGIN
                          TYPEIT(I,TAB,T1,TAB,(T2 = I*T1)) $,
                          NOLSTS = NOLSTS+T1 $,
                          IF (NOREFS = NOREFS+T2) GRT 100000
                          THEN BEGIN
                               REFSK = REFSK+1 $,
                               NOREFS = NOREFS-100000 $,
                               END $,
                          END $,
                     IF I EQL 5
                     THEN TYPEIT(LOCMES(.C. /TOTAL OF/),NOLSTS,LOCMES(.C. /LISTS AND/
                      ),REFSK,LOCMES(.C. /* 100000 +/),NOREFS,LOCMES(
                      .C. /REFERENCES,  UP HERE./)) $,
                     END $,
OUTINT $             IF COMARG(5) EQL -377777777777Q
                     THEN GOTO ENDEXC $,
                     TYPEIT(LOCMES(.C. /   /)) $,
                     TYPEIT(LOCMES(.C. /*******INTERVALS********/)) $,
                     TYPEIT(LOCMES(.C. / INTERVAL =/),INTVAL) $,
                     FOR I = 1 STEP 1 UNTIL IVMAX
                     DO   BEGIN
                          T3 = C(IVDIST.+I) $,
                          IF T3 NEQ 0
                          THEN TYPEIT((IVIDC = INTVAL*(I-1)+1),TAB,T3) $,
                          END $,
          COMMENT $,
ENDEXC $             DORMNT() $,
          COMMENT $,
OVRFLO $             TYPEIT(LOCMES(.C. /LIST TOO BIG  SIZE IS/),LSTSIZ) $,
                     DCOM(NAME.) = BYN(LIST.) $,
                     ADCOM(NAME.) = LIST.+3 $,
                     TYPEIT(LOCMES(.C. /LIST NAME IS/),NAME.) $,
                     GOTO BACK $,
OPNERR $             TYPEIT(LOCMES(.C. /ERROR IN OPENING/),BCDASC(N1),BCDASC(N2)) $,
                     DORMNT() $,
RDERR $              TYPEIT(LOCMES(.C. /ERROR IN READING/),BCDASC(N1),BCDASC(N2)) $,
                     DORMNT() $,
CLSERR $             TYPEIT(LOCMES(.C. /ERROR IN CLOSING/),BCDASC(N1),BCDASC(N2),LOCMES(
                      .C. /ERROR CODE IS/),(ERRCOD = RECODE())) $,
                     DORMNT() $,
                     END FINI
```

Appendix B

IFSTAT Progam Listing

```
BEGIN
COMMENT  ************************************************************************
  PROGRAM-NAME   IFSTAT
  COMMAND        R IFSTAT NAME1 NAME2 LSTLMT CUTREF
  FUNCTION       THIS PROGRAM GATHERS STATISTICS OF THE INVERTED FILE.  IT PRODUCES:
                 A.  A TABLE FOR REDUNDANCY MEASUREMENTS (NUMBER OF REFERENCES AND
                     NUMBER OF DISTINCT DOCUMENT REFERENCES, FOR EACH DOCUMENT
                     INTERVAL.)
                 B.  TABLES OF THE NUMBER OF STEM TYPES AS A FUNCTION
                     OF THE NUMBER OF ENDINGS, AND AS A FUNCTION OF THE STEM LENGTH.
                 C.  TABLES OF THE NUMBER OF REFERENCES AS A FUNCTION OF WORD NUMBERS,
                     AS A FUNCTION OF TERM NUMBERS, AND AS A FUNCTION OF RANGE NUMBERS.
  NOTE  1.       NAME1 SHOULD BE S1, NAME2 SHOULD BE THE LATEST SECONDARY NAME OF
                 THE INVERTED FILE.  LSTLMT DESIGNATES THE NUMBER OF LISTS (STEMS)
                 THIS PROGRAM PROCESSES.  IF IT EXCEEDS THE CURRENT NUMBER OF LISTS,
                 ALL LISTS ARE PROCESSED.  CUTREF SHOULD USUALLY BE 0 (ZERO).  IF
                 IT IS NOT ZERO, STEMS WITH FREQUENCY NOT GREATER THAN THAT NUMBER
                 WILL NOT BE PROCESSED.  (THIS IS USED TO GATHER STATISTICS FOR HIGH
                 FREQUENCY STEMS.
         2.      STATISTICS FOR THE NUMBER OF ENDINGS DISTRIBUTION SHOULD BE COMPARED
                 WITH THAT OBTAINED FROM LSTDST.  BOTH SHOULD BE IDENTICAL.  (TWO
                 PROGRAMS TRACE THE INVERTED FILE IN DIFFERENT WAYS.)
************************************************************************************
 $,
            .INSERT IFLCOM $,  ... PACKED COMPONENTS//
            CONT .BCD. .CONT. $,
            INTEGER SIZSEC,BUFIN,NWORDS,EOFCNT,NBNAME,BP,NAM1,NAM2,FILNO,OFFSET,NAMDUM,BLANKS,INPTR,ENDAD,REFS,
            NOENDS,NAMLEN,DEPTH,PRE,NAMWD1,PART2,LSTCNT,LSTLMT,PRVREF,CURREF,CUR100,REFPTR,DCEPTR,REFSID,CUTREF,.
            I,J,MAXSIZ,REFITM,DCEITM,RRR,TAB,DCLDG,DCLOR,EDSPTR,EDSITM,CURMAX,EDSMAX,BYNMAX,BYNPTR,WNPTR,TNPTR,
            WGTPTR,NOCOM $,
            INTEGER ARRAY BUF1(432),BUF2(432),SECT(432) $,
            INTEGER PROCEDURE COMARG,LOCMES,FREZ,FRET,BCDEC,FILCNT,DEFBC,.C.ASC  $,
            PROCEDURE BEOPEN,BECLOS,TYPEIT,CHNCOM,UPNAM,DELIST,BEREAD,RRRCNT,PRNTAL $,
            BOOLEAN READIN,SWEOF  $,
            PRESET BEGIN
                              SIZSEC=432 $,
                              SWEOF=0 $,
                              LSTCNT=0 $,
                              CURMAX=300 $,
                              EDSMAX=30 $,
                              BYNMAX=64 $,

                   END $,
            REFPTR=FREZ(CURMAX+1) $,
            DCEPTR=FREZ(CURMAX+1) $,
            EDSPTR=FREZ(EDSMAX+1) $,
            BYNPTR=FREZ(BYNMAX+1) $,
            WNPTR=FREZ(17) $,
            TNPTR=FREZ(65) $,
            WGTPTR=FREZ(9) $,
            TAB=.C.ASC(.C. /   /) $,
            NAMDUM=COMARG(1)$,
            NAM2=COMARG(2) $,
            LSTLMT=BCDEC(COMARG(3)) $,
            CUTREF=BCDEC(COMARG(4)) $,
            FILNO=0$,
            UPNAM() $,
OPENIT $    BEOPEN(.BCD. /R/,NAM1,NAM2,BUF1,BUF2,-0,RERR) $,
QUEST $     BEREAD(NAM1,NAM2,SECT TO SIZSEC,SETEND,EOFCNT,RERR) $,
            NWORDS=SIZSEC $,
FULL $      BUFIN=LOC SECT $,
            DEPTH=0 $,
            IF LEFTHF(BUFIN) NEQ 777777C THEN GOTO FENERR $,
            OFFSET=ADCOM(BUFIN) $,        ... GET SECTION LIST POINTER//
            IF OFFSET EQL 0 THEN GOTO OFFERR $,
            BUFIN=BUFIN+OFFSET-1 $,
NEXLIS $    IF HEDR(BUFIN) NEQ 7 THEN GOTO HEDERR $,          ... LIST HEADER MISSING//
            LSTCNT=LSTCNT+1 $,
            IF LSTCNT GRT LSTLMT THEN
                 BEGIN
                              LSTCNT=LSTLMT $,
                              GOTO PRNT $,
                              END $,
            BLANKS=BWL(BUFIN) $,
            NBNAME=BYN(BUFIN) $,
            INPTR=BUFIN+4 .V. (NBNAME .LS. 18) $,
            NAMWD1=W(INPTR) $,    ... SAVE 1ST 4 LETTERS OF NAME //
            PRE=0 $,
            DCLDG=0 $,
            DCLOR=DCL(BUFIN) $,
            PRVREF=999999 $,
            NOCOM=BYN(BUFIN) $,
            IF NOCOM GRT 64  THEN
                              TYPEIT(LOCMES(.C. /BYN TOO BIG --/),NOCOM) $,
            W(BYNPTR+NOCOM)=W(BYNPTR+NOCOM)+1 $,
            NOENDS=EDS(BUFIN) $,          ... NO. OF ENDING STRINGS//
```

```
            W(EDSPTR+NOENDS)=W(EDSPTR+NOENDS)+1 $,
            NAMLEN=(NBNAME+3)/4 $,
            BP=4+NAMLEN+NOENDS*2 $,       ... PUT PTR AT 1ST REF.//
PREREF $    REFS=REL(BUFIN) $,
            REFSID=REL(BUFIN) $,
CUTAFX $    ENDAD=BUFIN+4+NAMLEN $,
            PART2=(ENDAD+NOENDS*2)-(LOC SECT+SIZSEC) $,
            IF PART2 GRT 0 THEN
                              BEGIN
                              PRE=PART2+1 $,
                              BEREAD(NAM1,NAM2,SECT TO PRE,OEEND,EOECNT,RERR) $,
                              END $,
DOREFS $    IF BP+DEPTH GEQ SIZSEC THEN OFLIST() $,   ... GET NEXT SEC. //
            RBRCNT() $,
UPBP$       BP=BP+1$,
            IF (REFS=REFS-1) EQL 0 THEN GOTO MORE $,
            GOTO DOREFS$,
SETEND $    SWEOF=TRUE $,
            GOTO MORE $,
MORE $      BUFIN=BUFIN+BP-1+BLANKS $,    ... ADJUST INPUT POINTER//
            IF DCLOR NEQ DCLDG
            THEN BEGIN
                 TYPEIT(LOCMES(.C. /CHECK THIS DCL/),DCLOR,DCLDG) $,
                 TYPEIT(LOCMES(.C. /IN FILE NO. -/),FILNO) $,
                 END $,
            DEPTH=BUFIN-LOC SECT $,       ... DISTANCE INTO SECTION//
            IF DEPTH LES SIZSEC-1 THEN GOTO NEXLIS $,
            IF SWEOF THEN GOTO FINIS $,
            GOTO QUEST $,       ... READ MORE INPUT//
DEFINE PROCEDURE OFLIST() TOBE
            BEGIN
            BEREAD(NAM1,NAM2,SECT(PRE) TO SIZSEC-PRE,OEEND,EOECNT,RERR) $,
FULL2 $     NWORDS=SIZSEC $,
            IF PRE NEQ 0 THEN BP=PRE ELSE
            BP=BP+DEPTH-SIZSEC+1 $,
            BUFIN=LOC SECT $,
            PRE=DEPTH=0 $,
            END $,
FINIS $     RECLOS(NAM1,NAM2,KEY) $,
            SWEOF=FALSE $,
            UPNAM() $,
            NWORDS=FILCNT(NAM1,NAM2) $,
OUT $       IF NWORDS GRT 0 THEN GOTO OPENIT $,
PRNT $      PRNTAL() $,
            CHNCOM() $,
OEEND $     IF EOECNT EQL 0 THEN GOTO MORE $,
            TYPEIT(LOCMES(.C. /PREMATURE END OF FILE REACHED/)) $,
            GOTO SHOW2 $,
KEY $       NWORDS=0$,
            GOTO OUT $,
RERR $      TYPEIT(LOCMES(.C. /READ ERROR/)) $,
            GOTO SHOWER $,
FENERR $    TYPEIT(LOCMES(.C. /FENCE ERROR/)) $,
            GOTO SHOWER $,
NAMERR $    TYPEIT(LOCMES(.C. /LIST NAME ERROR./)) $,
            GOTO SHOWER $,
DCLERR $    TYPEIT(DCLOR,DCLDG) $,
                              GOTO SHOWER $,
OFFERR $    TYPEIT(LOCMES(.C. /OFFSET ERROR/)) $,
                              GOTO SHOWER $,
HEDERR $    TYPEIT(LOCMES(.C. /LIST HEADER ERROR/)) $,
SHOWER $    INPTR=LOC NAMWD1 .V. 4C18 $,
            TYPEIT(LOCMES(.C. /LAST LIST NAME CONTAINED---/),INPTR) $,
SHOW2 $     TYPEIT(LOCMES(.C. /IN FILE NO.-/),FILNO) $,
            GOTO KEY $,
DEFINE PROCEDURE UPNAM() TOBE
            BEGIN
            FILNO=FILNO+1 $,
            LEFTHE(LOC NAM1)=NAMDUM $,
            RITEHE(LOC NAM1)=DEERC(FILNO) $,
            END $,
DEFINE PROCEDURE RBRCNT() TOBE
            BEGIN
            IF REFSID LEQ CUTREF THEN GOTO RETURN $,
            NOCOM=WRDNO(BUFIN+BP) $,
            W(WNPTR+NOCOM)=W(WNPTR+NOCOM)+1 $,
            NOCOM=DOCPOS(BUFIN+BP)$,
            W(TNPTR+NOCOM)=W(TNPTR+NOCOM)+1 $,
            NOCOM=WGTCOM(BUFIN+BP) $,
            W(WGTPTR+NOCOM)=W(WGTPTR+NOCOM)+1 $,
            CURREF=ADCOM(BUFIN+BP) $,
            CUR100=(CURREF-1)/1000+1 $,
            IF CUR100 GRT CURMAX
                              THEN BEGIN
                                   TYPEIT(LOCMES(.C. /REFERENCE TABLE OVERFLOW .../), CUR100) $,
                                   CHNCOM() $,
                                   END $,
            W(REFPTR+CUR100)=W(REFPTR+CUR100)+1 $,
            IF PRVREF NEQ CURREF THEN
                              BEGIN
                              PRVREF=CURREF $,
```

```
            DCLDG=DCLDG+1 $,
                              W(DCEPTR+CUR100)=W(DCEPTR+CUR100)+1 $,
                              END $,
            END $,
DEFINE PROCEDURE PRNTAL () TORE
            BEGIN
            REEITM=0 $,
            DCEITM=0 $,
            TYPEIT(LSTCNT,LOCMES(.C. /LISTS PROCESSED./)) $,
            TYPEIT(LOCMES(.C. /    /)) $,
            TYPEIT(LOCMES(.C. ,      DOC.NO       NO.REFS.   NO.DD.   R*100/D,)) $,
            FOR I=1 STEP 1 UNTIL CURMAX
                              DO BEGIN
                                        REEITM=W(REEPTR+I) $,
                                        IF REEITM NEQ 0
                                        THEN BEGIN
                                   IF I LEQ 10
                                   THEN  IF I EQL 10 THEN TYPEIT(LOCMES(.C. /  /),CONT)
                                        ELSE  IF I EQL 1 THEN TYPEIT(LOCMES(.C. /      /),CONT)
                                                         ELSE TYPEIT(LOCMES(.C. /   /),CONT)
                                   ELSE TYPEIT(LOCMES(.C. / /),CONT) $,
                              TYPEIT((J=1000*(I-1)+1), LOCMES(.C. /-/),(J=I*1000),
                              TAB,REEITM,TAB,(DCEITM=W(DCEPTR+I)),TAB,
                              (RRR=REEITM*100/DCEITM)) $,
                              END $,
            END $,
            TYPEIT(LOCMES(.C. /    /)) $,
            TYPEIT(LOCMES(.C. /NO.ENDS.    NO.LISTS  /)) $,
            FOR I=0 STEP 1 UNTIL EDSMAX
            DO BEGIN
               EDSITM=W(EDSPTR+I) $,
               IF EDSITM NEQ 0
               THEN TYPEIT(TAB,I,TAB,TAB,EDSITM) $,
               END $,
            TYPEIT(LOCMES(.C. /    /)) $,
            TYPEIT(LOCMES(.C. /STEM LGTH    NO.STEMS/)) $,
            FOR I=0 STEP 1 UNTIL BYNMAX
            DO BEGIN
               EDSITM=W(BYNPTR+I) $,
               IF EDSITM NEQ 0
               THEN TYPEIT(TAB,I,TAB,TAB,EDSITM) $,
               END $,
            TYPEIT(LOCMES(.C. /    /)) $,
            TYPEIT(LOCMES(.C. /WORD NO.    NO.REFS./)) $,
            FOR I=0 STEP 1 UNTIL 16
            DO BEGIN
               EDSITM=W(WNPTR+I) $,
               IF EDSITM NEQ 0
               THEN TYPEIT(TAB,I,TAB,TAB,EDSITM) $,
               END $,
            TYPEIT(LOCMES(.C. /    /)) $,
            TYPEIT(LOCMES(.C. /TERM NO.  NO.REFS./)) $,
            FOR I=0 STEP 1 UNTIL 64
            DO BEGIN
               EDSITM=W(TNPTR+I) $,
               IF EDSITM NEQ 0
               THEN TYPEIT(TAB,I,TAB,TAB,EDSITM) $,
               END $,
            TYPEIT(LOCMES(.C. /    /)) $,
            TYPEIT(LOCMES(.C. /RANGE NO.  NO.REFS./)) $,
            FOR I=0 STEP 1 UNTIL 8
            DO BEGIN
               J=0 $,
               EDSITM=W(WGTPTR+I) $,
               IF EDSITM NEQ 0
               THEN BEGIN
     LOOP00 $  IF EDSITM GRT 100000
               THEN BEGIN
                    J=J+1 $,
                    EDSITM=EDSITM-100000 $,
                    GOTO LOOP00 $,
                    END $,
               TYPEIT(TAB,I,TAB,TAB,J,LOCMES(.C. / * 100000 +/),
                      EDSITM) $,
               END $,
               END $,
END $,
END FINI
```

Appendix C

IFDOCS Program Listing

```
BEGIN
COMMENT  ***************************************************************************
  PROGRAM-NAME   IFDOCS
  COMMAND        R  IFDOCS  NAME1  NAME2
  FUNCTION       THIS PROGRAM PRODUCES A LIST OF DOCUMENT NUMBERS THAT HAVE NOT
                 YET BEEN USED (UP TO DOC. NO. 18000) AND DOCUMENT NUMBERS
                 THAT HAVE ALREADY BEEN USED (DOC. NO. 18001 - 36000.)
  NOTE  1.       NAME1 SHOULD BE SI, NAME2 SHOULD BE THE LATEST SECONDARY NAME
                 OF THE INVERTED FILE.
        2.       FOR SAVING PRINTING VOLUMES, YOU CAN ALTER THE BORDER NUMBER
                 (NOW 18000) BY CHANGING MAXDOC.  HOWEVER IT WILL NOT BE ABLE TO
                 EXCEED 21000 BECAUSE OF FREE STORAGE SHORTAGE.
***************************************************************************
 $,
          .INSERT IELCOM $,  ... PACKED COMPONENTS//
          CONT .BCD. .CONT. $,
          INTEGER SIZSEC,BUFIN,NWORDS,EOFCNT,NBNAME,BP,NAM1,NAM2,FILNO,OFFSET,NAMDUM,BLANKS,INPTR,ENDAD,REFS,
          NOENDS,NAMLEN,DEPTH,PRE,NAMWD1,PART2,LSTCNT,CURREF,MAXDOC,NOTAB.,REFEXT,
          I,J,K,MAXSIZ,REFITM,DOCITM,RPR,TAB  $,
          INTEGER ARRAY BUF1(432),BUF2(432),SECT(432) $,
          INTEGER PROCEDURE COMARG,LOCMES,FREZ,FPET,BCDEC,FILCNT,DEFBC,.C.ASC  $,
          PROCEDURE BEOPEN,BECLOS,TYPEIT,CUNCOM,UPNAM,OFLIST,BEREAD,RRRCNT,PRNTAL $,
          BOOLEAN READIN,SWEOF  $,
          PRESET BEGIN
                              SIZSEC=432 $,
                              SWEOF=0 $,
                              LSTCNT=0 $,
                              MAXDOC=18000 $,       ... MAXIMUM DOCUMENT NUMBER //
                 END $,
          TAB=.C.ASC(.C. /   /) $,
          NAMDUM=COMARG(1)$,
          NAM2=COMARG(2) $,
          NOTAB.=FREZ(MAXDOC+1) $,
          FOR I=1 STEP 1 UNTIL MAXDOC
          DO BEGIN
             W(NOTAB.+I)=0 $,       ... INITIALIZE W-TABLE//
             END $,
          FILNO=0$,
          UPNAM() $,
OPENIT $  BEOPEN(.BCD. /R/,NAM1,NAM2,BUF1,BUF2,-0,RERR) $,
QUEST $   BEREAD(NAM1,NAM2,SECT TO SIZSEC,SETEND,EOFCNT,RERR) $,
          NWORDS=SIZSEC $,
FULL $    BUFIN=LOC SECT $,
          DEPTH=0 $,
          IF LEFTHF(BUFIN) NEQ 777777C THEN GOTO FENERR $,
          OFFSET=ADCOM(BUFIN) $,        ... GET SECTION LIST POINTER//
          BUFIN=BUFIN+OFFSET-1 $,
NEXLIS $  IF HEDR(BUFIN) NEQ 7 THEN GOTO HEDERR $,          ... LIST HEADER MISSING//
          LSTCNT=LSTCNT+1 $,
          BLANKS=RWL(BUFIN) $,
          NBNAME=RYN(BUFIN) $,
          INPTR=BUFIN+4 .V. (NBNAME .LS. 18) $,
          NAMWD1=W(INPTR) $,    ... SAVE 1ST 4 LETTERS OF NAME //
          PRE=0 $,
          NOENDS=EDS(BUFIN) $,          ... NO. OF ENDING STRINGS//
          NAMLEN=(NBNAME+3)/4 $,
          BP=4+NAMLEN+NOENDS*2 $,       ... PUT PTR AT 1ST REF.//
PREREF $  REFS=REL(BUFIN) $,
OUTAFX $  ENDAD=BUFIN+4+NAMLEN $,
          PART2=(ENDAD+NOENDS*2)-(LOC SECT+SIZSEC) $,
          IF PART2 GRT 0 THEN
                           BEGIN
                           PRE=PART2+1 $,
                           BEREAD(NAM1,NAM2,SECT TO PRE,OFEND,EOFCNT,RERR) $,
                           END $,
DOREFS $  IF BP+DEPTH GEQ SIZSEC THEN OFLIST() $,    ... GET NEXT SEC. //
          CURREF=ADCOM(BUFIN+BP) $,
          IF CURREF GRT MAXDOC
          THEN BEGIN
               CURREF=CURREF-MAXDOC $,
               DECCOM(NOTAB.+CURREF)=1 $,
               END
          ELSE ADCOM(NOTAB.+CURREF)=1 $,
UPBP$     BP=BP+1$,
          IF (REFS=REFS-1) EQL 0 THEN GOTO MORE $,
          GOTO DOREFS$,
SETEND $  SWEOF=TRUE $,
          GOTO MORE $,
MORE $    BUFIN=BUFIN+BP-1+BLANKS $,    ... ADJUST INPUT POINTER//
          DEPTH=BUFIN-LOC SECT $,       ... DISTANCE INTO SECTION//
          IF DEPTH LES SIZSEC-1 THEN GOTO NEXLIS $,
          IF SWEOF THEN GOTO FINIS $,
          GOTO QUEST $,       ... READ MORE INPUT//
```

```
DEFINE PROCEDURE DFLIST() TOBE
            BEGIN
            BEREAD(NAM1,NAM2,SECT(PRE) TO SIZSEC-PRE,DFEND,EOFCNT,RERR) $,
FULL2 $     NWORDS=SIZSEC $,
            IF PRE NEQ 0 THEN BP=PRE ELSE
            BP=BP+DEPTH-SIZSEC+1 $,
            BUFIN=LOC SECT $,
            PRE=DEPTH=0 $,
            END $,
FINIS $     BECLOS(NAM1,NAM2,KEY) $,
            SWEOF=FALSE $,
            UPNAM() $,
            NWORDS=FILCNT(NAM1,NAM2) $,
OUT $       IF NWORDS GRT 0 THEN GOTO OPENIT $,
            TYPEIT(LSTCNT, LOCMES(.C. /LISTS PROCESSED. /)) $,
PRNT $      PRNTAL() $,
            CHNCOM() $,
DFEND $     IF EOFCNT EQL 0 THEN GOTO MORE $,
            TYPEIT(LOCMES(.C. /PREMATURE END OF FILE REACHED/)) $,
            GOTO SHOW2 $,
KEY $       NWORDS=0$,
            GOTO OUT $,
RERR $      TYPEIT(LOCMES(.C. /READ ERROR/)) $,
            GOTO SHOWER $,
FENERR $    TYPEIT(LOCMES(.C. /FENCE ERROR/)) $,
            GOTO SHOWER $,
NAMERR $    TYPEIT(LOCMES(.C. /LIST NAME ERROR./)) $,
            GOTO SHOWER $,
HEDERR $    TYPEIT(LOCMES(.C. /LIST HEADER ERROR/)) $,
SHOWER $    INPTR=LOC NAMWD1 .V. 4C18 $,
            TYPEIT(LOCMES(.C. /LAST LIST NAME CONTAINED---/),INPTR) $,
SHOW2 $     TYPEIT(LOCMES(.C. /IN FILE NO.-/),FILNO) $,
            GOTO KEY $,
DEFINE PROCEDURE UPNAM() TOBE
            BEGIN
            FILNO=FILNO+1 $,
            LEFTHF(LOC NAM1)=NAMDUM $,
            RITEHF(LOC NAM1)=DEFBC(FILNO) $,
            END $,
DEFINE PROCEDURE PRNTAL () TOBE
            BEGIN
            TYPEIT(LOCMES(.C. /  /)) $,
            TYPEIT(LOCMES(.C. /** (UP TO DOC. NO. 18000) NO REFERENCES TO THOSE DOCUMENTS **/)) $,
            J=0 $,
            FOR I=1 STEP 1 UNTIL MAXDOC DO
            BEGIN
                        IF (REFEXT=ADCOM(NOTAB,+I)) EQL 0
                        THEN BEGIN
                  TYPEIT(I, CONT) $,
                                    J=J+1 $,
                                    IF J EQL 10 THEN
                                    BEGIN
                                    TYPEIT(LOCMES(.C. / /)) $,
                                    J=0 $,
                                    END $,
                                    END $,
            END $,
            TYPEIT(LOCMES(.C. /    /)) $,
            TYPEIT(LOCMES(.C. /** (OVER DOC.NO. 18000) THOSE DOCUMENT NUMBERS REFERRED **/)) $,
            J=0 $,
            FOR I=1 STEP 1 UNTIL MAXDOC DO
            BEGIN
            IF (REFEXT=DECCOM(NOTAB,+I)) NEQ 0
            THEN BEGIN
                 TYPEIT((K=I+MAXDOC), CONT) $,
                 J=J+1 $,
                 IF J EQL 0 THEN
                 BEGIN
                 TYPEIT(LOCMES(.C. /    /)) $,
                 J=0 $,
                 END $,
                 END $,
            END$,
                                    END $,
END FINI
```

Appendix D

MIFLST Program Listing

```
BEGIN
COMMENT  ***************************************************************************
  PROGRAM-NAME  MIFLST
  COMMAND       R  MIFLST  NAME1  NAME2  GLSPEC  NUMBER  -MODE-  -SEGMENT-
  FUNCTION      MIFLST PRODUCES AN ASCII PRINT FILE WHICH WILL LIST A PART OF THE
                CONTENTS OF THE SUBJECT/TITLE INVERTED FILE.  THE RANGE OF
                LISTING CAN BE DESIGNATED BY 'GLSPEC NUMBER' COMBINATION.
  NOTE  1.      GLSPEC SHOULD BE 'G' OR 'L'.  NUMBER CAN BE ANY INTEGER.  THEY SPECIFY
                THE RANGE OF OUTPUT LISTING.  E.G.
                  'G  1000'  MEANS GREATER THAN OR EQUAL TO 1000.  IF SO SPECIFIED,
                THIS PROGRAM PRODUCES A LIST OF STEMS WITH FREQUENCY MORE
                THAN 999.
                  'L  5'  MEANS LESS THAN OR EQUAL TO 5.  IN THIS CASE, THOSE STEMS
                WITH FREQUENCY LESS THAN 6 WILL BE PRINTED OUT.
        2.      FOR OTHER ARGUMENTS, SEE THE FOLLOWING COMMENTS.
***************************************************************************
IF MODE IS 0 OR ABSENT A COMPLETE LIST IS CREATED. IF MODE IS 1 NO REFERENCES ARE LISTED.
IF A SEGMENT NO. IS GIVEN (6TH ARG.), ONLY THAT SEGMENT WILL BE LISTED. IF A FULL FILE
FIRST NAME (E.G. SI001) IS GIVEN, THEN ALL EXISTING FILES FROM THAT POINT ON WILL BE LISTED.
IF ONLY THE PREFIX SI IS GIVEN AS NAME1 THEN ALL FILES WILL BE LISTED UNLESS A SEGMENT
NO. IS GIVEN IN ARG. 6. IF NAME1 DOES NOT CONTAIN THE PREFIX SI, THEN A BATCH FILE OF
NAME1 NAME2 CONTAINING NAME PAIRS IS USED TO SUPPLY THE NAMES OF FILES TO BE LISTED.
THE OUTPUT FILES WILL HAVE THE SAME NAME1 AS THE BATCH FILE IN BATCH MODE.
IN NON-BATCH MODE, NAME1 WILL BE SUBFIL. NAME2 IS ALWAYS IFLIST. $,
     .INSERT IFLCOM $,  ... PACKED COMPONENTS//
     IFLIST .BCD. /IFLIST/ $,
     ENDING .BCD. /ENDING/ $,
     TEST1  .BCD. / TEST1/ $,
     SUBFIL .BCD. /SUBFIL/ $,
     SUBDUM .BCD. / SI/ $,
    LEOPAR .BCD. /     L/$,
    GEOPAR .BCD. /     G/$,
     INTEGER SIZSEC,REFLIN,COLUM2,BUFIN,BUFOUT,NWORDS,EOFCNT,NBNAME,NBSTEM,I,K,N,MODE,BP,NAM1,NAM2,NAMPTR,
     BYTES,LEN,M,NULL,SETHED,TOTCOD,OUTMOD,SIZBUF,DELS,DELPTR,DELF,FILNO,OFFSET,NAMDUM,TOTWRD,BLANKS,TEMP,
     REL,ASCPTR,HEDPTR,INPTR,ENDAD,ECODS,REFS,NOENDS,NAMLEN,ENDGS,MODE,DEPTH,ONAM1,PRE,BATCH2,NAMWD1,CODES,
    CUTGRT,CUTLES,CPRPAR,DOCNO $,
     INTEGER ARRAY BUF1(432),BUF2(432),SECT(432),WBUF1(432),WBUF2(432),WBUF3(432),OUTBUF(864),HEADER(15) $,
     INTEGER PROCEDURE COMARG,ASCII6,INTASC,GETEND,GROUP,GET,LOCMES,FREZ,FRET,NEXITM,BCDEC,FILCNT,DEFBC $,
     PROCEDURE REOPEN,RECLOS,REWRIT,TYPEIT,CHNCOM,COPY,PUT,INC,CUTREF,COLNUM,UPNAM,OFLIST,BFREAD,
     TABIN,RITOUT,DEC1 $,
     BOOLEAN PROCEDURE SETEXT,XTRACT $,
     BOOLEAN NOREFS,READIN,SEG,SWEOF,EOSONE,BATCH,OUTRNG $,
     PRESET BEGIN
                        NULL=0 $,
                        DELS=040055052C9 $, ... SPACE, HYPHEN, ASTER.//
                        NOREFS=0 $,
     OUTRNG=0 $,
                        READIN=0 $,
                        SEG=0 $,
                        TOTCOD=0 $,
                        SIZBUF=864 $,
                        SIZSEC=432 $,
                        REFLIN=5 $,
                        COLUM2=30 $,
                        CUTGRT = 0$,
                        CUTLES = 999999$,
          END $,
     PRESET SETHED=0120110110110 $,
     PRESET HEADER=012012040040C,040040124145C,1621550110110,040040040040C,040116157056C,040122145146C,16304004011
     157056040104C,157143163040C,040116157056C,040105156144C,1630120120000 $,       ... COLUMN HEADINGS IN ASCII//
     HEDPTR=LOC HEADER .V. 57C18 $,
     DELPTR=LOC DELS .V. 200003C18 $,
     BATCH=SWEOF=FALSE $,
     FILNO=0 $,
     NAMDUM=COMARG(1) $,           ... ACCEPT FILE NAMES FROM CONSOLE//
     NAM2=COMARG(2) $,
     CPRPAR = COMARG(3)$,
     IF CPRPAR EQL LEOPAR
                        THEN CUTGRT = BCDEC(COMARG(4))
                        ELSE BEGIN
                        IF CPRPAR EQL GEOPAR
                        THEN CUTLES = BCDEC(COMARG(4))
                        ELSE BEGIN
                                  TYPEIT(LOCMES(.C. /** GLSPEC ERROR. SHOULD BE 'G' OR 'L'./))$,
                                  CHNCOM()$,
                                  END$,
                        END$,
CHKNAM $   IF LEFTHF(LOC NAMDUM) EQL LEFTHF(LOC SUBDUM) THEN
                        BEGIN
                        FILNO=BCDEC(NAMDUM .A. 777777C)-1 $,     ... GET STARTING FILE NO.//
                        NAMDUM=NAMDUM .RS. 18 $,
                        GOTO SF $,
                        END $,
     IF RITHF(LOC NAMDUM) EQL LEFTHF(LOC SUBDUM) THEN GOTO SF $,
     BATCH=TRUE $,
```

```
            ONAM1=NAMDUM $,
            BATCH2=NAM2 $,
            IF SETEXT(NAMDUM,BATCH2) THEN GOTO NOBAT $,
            SF $                    ONAM1=SUBFIL $,
MOD $       IF (OUTMOD=COMARG(5)) EQL -377777777777C THEN GOTO HEADIT $,          ... MODE 0 IMPLIED//
            OUTMOD=BCDEC(OUTMOD) $,        ... CONVERT NO. TO BINARY//
            IF OUTMOD NEQ 0 THEN NOREES=TRUE $,
            IF(N=COMARG(6)) NEQ -377777777777C THEN
                                    BEGIN
                                    SEG=TRUE $,
                                    FILNO=BCDEC(N) $,    ... CONVERT FILE NO. TO BINARY//
                                    FILNO=FILNO-1 $,
                                    END $,
HEADIT $    IF BATCH THEN
                                    BEGIN
                                    IF XTRACT(NAM1,NAM2) THEN GOTO NOBAT $,
                                    GOTO HEDNEW $,
                                    END $,
            UPNAM() $,              ... GENERATE NAME 1//
HEDNEW $    BUFOUT=LOC OUTBUF $,
            ASCPTR=LOC SETHED .V. 4C18 $,
            COPY(ASCPTR,BUFOUT) $,          ... INSERT C. R. AND 3 TABS//
            INC(BUFOUT,4) $,
            NAMPTR=LOC NAM1 .V. 6C18 $,
            ASCPTR=ASCIT6(NAMPTR) $,
            COPY(ASCPTR,BUFOUT) $,          ... INSERT NAME1 OF FILE//
            INC(BUFOUT,6) $,
            PUT(40C,BUFOUT) $,
            INC(BUFOUT,1) $,
            NAMPTR=LOC NAM2 $,
            NAMPTR=NAMPTR .V. 6C18 $,
            ASCPTR=ASCIT6(NAMPTR) $,
            COPY(ASCPTR,BUFOUT) $,          ... INSERT NAME2 OF FILE//
            INC(BUFOUT,6) $,
            COPY(HEDPTR,BUFOUT) $,          ... INSERT COLUMN HEADINGS//
            INC(BUFOUT,47) $,
            BEOPEN(.BCD. /W/,ONAM1,IELIST,WBUF1,WBUF2,WBUF3,WERR) $,   ... OPEN OUTPUT FILE//
OPENIT $    BEOPEN(.BCD. /R/,NAM1,NAM2,BUF1,BUF2,-0,RERR) $,
QUEST $     BEREAD(NAM1,NAM2,SECT TO SIZSEC,SETEND,EOECNT,RERR) $,
            NWORDS=SIZSEC $,
FULL $      BUFIN=LOC SECT $,
            DEPTH=0 $,
            IF LEFTHE(BUFIN) NEQ 777777C THEN GOTO FENERR $,
            OFFSET=ADCOM(BUFIN) $,          ... GET SECTION LIST POINTER//
            IF OFFSET EQL 0 THEN GOTO OFFERR $,
            BUFIN=BUFIN+OFFSET-1 $,
NEXLIS $    IF HEDR(BUFIN) NEQ 7 THEN GOTO HEDERR $,          ... LIST HEADER MISSING//
            BLANKS=BWL(BUFIN) $,
            NBNAME=NBSTEM=BYN(BUFIN) $,
            INPTR=BUFIN+4 .V. (NBNAME .LS. 18) $,  ... CREATE POINTER TO NAME STRING//
            NAMWD1=W(INPTR) $, ... SAVE 1ST 4 LETTERS OF NAME//
            NAMPTR=INPTR $,
            IF CODES NEQ 0 THEN FRET(TOTCOD,CODES) $,
            IF TEMP NEQ 0 THEN FRET(TOTWRD,TEMP) $,
            TEMP=CODES=0 $,
            PRE=ENDGS=0 $,
            NOENDS=EDS(BUFIN) $,            ... NO. OF ENDING STRINGS//
            NAMLEN=(NBNAME+3)/4 $,
            BP=4+NAMLEN+NOENDS*2 $,         ... PUT PTR AT 1ST REF.//
            REFS = REL(BUFIN) $,
            IF REFS LES CUTLES AND REFS GRT CUTGRT
            THEN BEGIN
                                OUTRNG=TRUE $,
                                GOTO PREREF $,
                                END $,
            OUTRNG=FALSE $,
            ENDAD=BUFIN+4+NAMLEN $,         ... COMPUTE ADDRESS OF ENDING STRING 1//
            ECODS=GROUP(ENDAD,NOENDS) $, ... MAKE LIST OF ENDING CODES//
            ECODS=ADCOM(LOC ECODS) $,
TESTED $    IF NOENDS EQL 1 THEN GOTO JOIN $,       ... ONLY 1 ENDING STRING//
            COPY(INPTR,BUFOUT) $,           ... INSERT STEM//
            INC(BUFOUT,BYN(BUFIN)) $,
            PUT(55C,BUFOUT) $, ... INSERT HYPHEN//
            INC(BUFOUT,1) $,
            NBNAME=NBNAME+1 $,
COL2 $      TABIN(NBNAME) $,   ... TAB TO COLUMN TWO//
            COLNUM(REL(BUFIN),DCL(BUFIN),NOENDS) $,          ... COLUMNIZE COUNTS//
LINE2 $     PUT(12C,BUFOUT) $, ... END LINE ONE//
            INC(BUFOUT,1) $,
            K=0 $,
            IF EDSONE THEN BEGIN EDSONE=FALSE $, GOTO PREREF $, END $,
            FOR I=0 STEP 1 WHILE I LES NOENDS DO   ... LIST WHOLE NAMES IF MORE THAN 1 ENDING STRING//
                                BEGIN
                                 PUT(40C,BUFOUT) $, ... INDENT 1 SPACE//
                                INC(BUFOUT,1) $,
            COPNAM $            COPY(NAMPTR,BUFOUT) $,          ... INSERT THIS WORD (STEM)//
                                INC(BUFOUT,NBSTEM) $,
              APPEND $ ASCPTR=GETEND(W(ECODS+K)) $,          ... CONVERT ENDING CODE//
                                IF EDSONE THEN IF DECCOM(LOC ASCPTR) NEQ 0 THEN
                                BEGIN
                                PUT(55C,BUFOUT) $,  ... INSERT HYPHEN//
```

```
                    INC(BUFOUT,1) $,
                    ENDGS=ENDGS+1 $,
                    GOTO COP $,
                    END
                    ELSE GOTO COP ELSE GOTO COP $,
          LISTIN $  ASCPTR=(ENDAD+1) .V. (TAGCOM(ENDAD) .LS. 18) $, ... CREATE POINTER TO INITIALS//
                    PUT(54C,BUFOUT) $,  ... INSERT COMMA//
                    INC(BUFOUT,1) $,
                    ENDGS=ENDGS+1 $,
        COP $ COPY(ASCPTR,BUFOUT) $,        ... INSERT THIS ENDING//
                    INC(BUFOUT,DECCOM(LOC ASCPTR)) $,
                    ENDGS=ENDGS+DECCOM(LOC ASCPTR) $,       ... ACCUMULATE NO. OF ENDING CHARACTERS//
                    NBNAME=NBSTEM+ENDGS $,          ... ADJUST LENGTH OF NAME//
                    IF EDSONE THEN GOTO COL2 $,    ... GO GET COLUMN TWO COUNTS//
                    TABIN(NBNAME+1) $,  ... TAB TO COLUMN TWO OVER SPACE AND NAME//
                    COLNUM(REF(ENDAD),DCF(ENDAD)) $,        ... COLUMNIZE COUNTS//
                    DEC1(BUFOUT) $,     ... WIPE OUT TAB//
                    PUT(12C,BUFOUT) $,  ... INSERT C. R.//
                    INC(BUFOUT,1) $,
                    ENDAD=ENDAD+2 $,    ... MOVE TO NEXT ENDING STRING//
                    NAMPTR=INPTR $,
                    K=K+1 $,  ... UPDATE END-CODE COUNTER//
                    ENDGS=0 $,
                    END $,
PREREF $  IF BP+DEPTH GEQ SIZSEC AND NOT READIN THEN
                    BEGIN
                    OFLIST() $,         ... REFS START IN NEXT SECTION//
                    READIN=FALSE $,     ... ALLOW ADJUSTMENT OF BP AT SKIP//
                    END $,
          IF NOREFS THEN
          RT1 $     IF BP+REFS+DEPTH GRT NWORDS THEN
                    BEGIN
                    IF REFS EQL 0 THEN BP=BP-1 ELSE
                    BP=BP+REFS $,
                    REFS=0 $,
                    OFLIST() $,
                    IF ADCOM(LOC SECT) EQL 0 THEN BP=BP+1 $,
                    GOTO RT1 $,
                    END
                    ELSE GOTO TERMIN ELSE
          N=REFLIN $,   ... SET NO. OF  REFS. PER LINE//
DOREFS $  IF REFS GEQ N THEN
                     FOR I=1 STEP 1 UNTIL N DO      ... DO 1 FULL LINE OF REFS.//
                     BEGIN
                     IF BP+DEPTH GEQ SIZSEC THEN OFLIST() $, ... GET NEXT SECTION//
          IF OUTRNG THEN GOTO UPBP $,
                    CUTREF(BUFIN+BP) $, ... TAKE REF. APART//
                    IF ADCOM(LOC BUFOUT)-LOC OUTBUF GRT SIZBUF-50 THEN RITOUT() $, ... EMPTY BUFFER//
              UPBP $ BP=BP+1 $,
                     END
          ELSE GOTO FINREF $,       ... DO LAST (PARTIAL) LINE OF REFS.//
          IF (REFS=REFS-N) EQL 0 THEN GOTO TERMIN $,        ... ALL REFS. DONE//
          PUT(12C,BUFOUT) $, ... INSERT C. R.//
          INC(BUFOUT,1) $,
          GOTO DOREFS $,
FINREF $  N=REFS $,           ... SET NO. OF REFS. ON LAST LINE//
          GOTO DOREFS $,
JOIN $    EDSONE=TRUE $,      ... INDICATE ONLY 1 ENDING IN LIST//
          K=I=0 $,            ... SET UP LOOP TERMI.//
          GOTO COPNAM $,
SETEND $  SWEOF=TRUE $,
          GOTO WRIT $,
TERMIN $  IF OUTRNG THEN GOTO SKIP $,
                    PUT(12C,BUFOUT) $,  ... INSERT 2 C. R.'S BETWEEN LISTS//
          INC(BUFOUT,1) $,
SKIP $    IF NOREFS AND NOT READIN THEN BP=BP+REFS $,
WRITIT $  IF TAGCOM(LOC BUFOUT) NEQ 0 THEN        ... TEST IF LAST OUTPUT WORD FULL//
                   IF ADCOM(LOC BUFOUT)-LOC OUTBUF LES SIZBUF/2+30 THEN GOTO MORE ELSE GOTO WRIT
                   ELSE GOTO WRIT $,
WRIT $    RITOUT() $,   ... EMPTY BUFFER//
          GOTO MORE $,
DEFINE PROCEDURE RITOUT() TOBE
          BEGIN
          IF TAGCOM(LOC BUFOUT) EQL 0 THEN
                    GOTO HMUCH $,
          BYTES=4-TAGCOM(LOC BUFOUT) $,
          FOR M=0 STEP 1 WHILE M LES BYTES DO
                    BEGIN
                    PUT(NULL,BUFOUT) $,
                    INC(BUFOUT,1) $,
                    END $,
HMUCH $   LEN=ADCOM(LOC BUFOUT)-LOC OUTBUF $,     ... COMPUTE C/WORDS TO WRITE//
          IF LEN EQL 0 THEN GOTO RETURN $,        ... MUST HAVE JUST WRITTEN IT//
          REWRIT(ONAM1,IFLIST,OUTBUF TO LEN,WERR) $,
          BUFOUT=LOC OUTBUF $,          ... RESET OUTPUT POINTER//
          END $,
MORE $    BUFIN=BUFIN+BP-1+BLANKS $,    ... ADJUST INPUT POINTER//
          READIN=FALSE $,
          DEPTH=BUFIN-LOC SECT $,       ... DISTANCE INTO SECTION//
          IF DEPTH LES SIZSEC-1 THEN GOTO NEXLIS $,
          IF SWEOF THEN GOTO FINIS $,
```

```
            GOTO QUEST $,          ... READ MORE INPUT//
DEFINE PROCEDURE OFLIST() TOBE
            BEGIN
            BEREAD(NAM1,NAM2,SECT(PRE) TO SIZSEC-PRE,OFEND,EOFCNT,RERR) $,
FULL2 $     NWORDS=SIZSEC $,
            IF PRE NEQ 0 THEN BP=PRE ELSE
            BP=BP+DEPTH-SIZSEC+1 $,
            BUFIN=LOC SECT $,
            IF PRE EQL 0 THEN READIN=TRUE $,
            PRE=DEPTH=0 $,
            END $,
FINIS $     BECLOS(NAM1,NAM2,KEY) $,
            SWEOF=FALSE $,
            IF SEG THEN
                              BEGIN
            KEY $ NWORDS=0 $,
                              GOTO OUT $,
                              END $,
            IF BATCH THEN
                              BEGIN
                              IF XTRACT(NAM1,NAM2) THEN NWORDS=0 ELSE NWORDS=1 $, ... GET MORE NAMES IN BATCH FILE//
                              GOTO OUT $,
                              END $,
            UPNAM() $,
            NWORDS=FILCNT(NAM1,NAM2) $,
OUT $       BECLOS(ONAM1,IFLIST,WERR) $,
            IF NWORDS GRT 0 THEN GOTO HEDNEW $,
            CHNCOM() $,
OFEND $     IF EOFCNT EQL 0 THEN
            BEGIN
            SWEOF=TRUE $,
            GOTO TERMIN $,
            END $,
            TYPEIT(LOCMES(.C. /PREMATURE END OF FILE REACHED/)) $,
            GOTO SHOW2 $,
WERR $      TYPEIT(LOCMES(.C. /WRITE ERROR/)) $,
            GOTO SHOWER $,
RERR $      TYPEIT(LOCMES(.C. /READ ERROR/)) $,
            GOTO SHOWER $,
FENERR $    TYPEIT(LOCMES(.C. /FENCE ERROR/)) $,
            GOTO SHOWER $,
NAMERR $    TYPEIT(LOCMES(.C. /LIST NAME ERROR./)) $,
            GOTO SHOWER $,
HEDERR $    TYPEIT(LOCMES(.C. /LIST HEADER ERROR/)) $,
            GOTO SHOWER $,
OFFERR $    TYPEIT(LOCMES(.C. /OFFSET ERROR/)) $,
SHOWER $    INPTR=LOC NAMWD1 .V. 4C18 $,
            TYPEIT(LOCMES(.C. /LAST LIST NAME CONTAINED---/),INPTR) $,
SHOW2 $     TYPEIT(LOCMES(.C. /IN FILE NO.-/),FILNO) $,
            NWORDS=0 $,
            GOTO OUT $,
NOBAT $     TYPEIT(LOCMES(.C. /$BATCH FILE EMPTY OR NON-EXISTANT./)) $,
            CHNCOM() $,
DEFINE PROCEDURE UPNAM() TOBE
            BEGIN
            FILNO=FILNO+1 $,
            LEFTHF(LOC NAM1)=NAMDUM $,
            RITEHF(LOC NAM1)=DEFRC(FILNO) $,
            END $,
DEFINE PROCEDURE COLNUM(NUMS) WHERE INTEGER NUMS TOBE        ... ROUTINE TO COLUMNIZE NOS.//
            BEGIN
            INTEGER K,N $,
            INTEGER ARRAY ARGS(4) $,
            INTEGER PROCEDURE ISARGV $,
            K=1 $,
            ARGS=NUMS $,          ... TAKE 1ST NO.//
STORE $     IF (ARGS(K)=ISARGV(RETURN,(K+1),-1)) NEQ -1 THEN
                              BEGIN
                              K=K+1 $,
                              GOTO STORE $,
                              END $,
            N=K-1 $,
            FOR K=0 STEP 1 UNTIL N DO
                              BEGIN
                              ASCPTR=INTASC(ARGS(K)) $,      ... CONVERT NUMBER INTO ASCII//
                              COPY(ASCPTR,BUFOUT) $,         ... INSERT NO. INTO OUTPUT//
                              INC(BUFOUT,DECCOM(LOC ASCPTR)) $,
                              PUT(11C,BUFOUT) $,  ... INSERT TAB//
                              INC(BUFOUT,1) $,
                              END $,
            END $,
DEFINE PROCEDURE TABIN(LENGTH) WHERE INTEGER LENGTH TOBE     ... ROUTINE TO TAB PAST A CHARACTER STRING//
                                                                 ... TO 3RD TAB STOP (POSITION 30) OF LINE//
            BEGIN
            INTEGER TABS,K $,
            IF LENGTH GRT 29 THEN GOTO OVFLO $,     ... ALREADY PAST 3RD STOP//
            TABS=(COLUM2+9-LENGTH)/10 $, ... COMPUTE NO. OF TABS NEEDED//
  TABIT $ FOR K=1 STEP 1 UNTIL TABS DO
                              BEGIN
                              PUT(11C,BUFOUT) $,  ... INSERT TAB//
                              INC(BUFOUT,1) $,
                              END $,
```

```
          GOTO RETURN $,
OVFLO $   PUT(12C,BUFOUT) $, ... INSERT C. R. //
          INC(BUFOUT,1) $,
          TABS=3 $,          ... NEED ALL 3 TABS//
          GOTO TABIT $,
          END $,
DEFINE PROCEDURE CUTREE(REFAD) WHERE INTEGER REFAD TOBE     ... ROUTINE TO DISECT REF. WORDS//
          BEGIN
          INTEGER CYC1,K $,
          INTEGER ARRAY PYLET(3) $,
          PRESET PYLET=127C,112C,117C $,          ... PROPERTY CODE SYMBOLS--WJOP//
          ASCPTR=INTASC(ADCOM(REFAD)) $,          ... CONVERT DOC. NO. TO ASCII//
          COPY(ASCPTR,BUFOUT) $,
          INC(BUFOUT,DECCOM(LOC ASCPTR)) $,
          PUT(40C,BUFOUT) $,
          INC(BUFOUT,1) $,
          ASCPTR=INTASC(DOCPOS(REFAD)) $,         ... CONVERT DOC. POS. NO. TO ASCII//
          COPY(ASCPTR,BUFOUT) $,
          INC(BUFOUT,DECCOM(LOC ASCPTR)) $,
          PUT(40C,BUFOUT) $,
          INC(BUFOUT,1) $,
          ASCPTR=INTASC(WRDNO(REFAD)) $,          ... CONVERT WORD NO.//
          COPY(ASCPTR,BUFOUT) $,
          INC(BUFOUT,DECCOM(LOC ASCPTR)) $,
          PUT(40C,BUFOUT) $,
          INC(BUFOUT,1) $,
          ASCPTR=INTASC(ESNO(REFAD)) $,           ...,CONVERT NO. OF AFFIX STRINGS//
          COPY(ASCPTR,BUFOUT) $,
          INC(BUFOUT,DECCOM(LOC ASCPTR)) $,
          PUT(40C,BUFOUT) $,
          INC(BUFOUT,1) $,
          PUT(50C,BUFOUT) $, ... INSERT LEFT PAREN.//
          INC(BUFOUT,1) $,
          PUT((WGTCOM(REFAD) .V. 60C),BUFOUT) $, ... CONVERT WEIGHT TO ASCII//
          INC(BUFOUT,1) $,
          PUT(51C,BUFOUT) $, ... INSERT RIGHT PAREN.//
          INC(BUFOUT,1) $,
          K=1 $,              ... PREVENT -0 IN BIT TEST BELOW//
          PROPY $             FOR CYC1=0 STEP 1 UNTIL 2 DO
                              BEGIN
              PC $  BYTE1(LOC K)=PRYCOM(REFAD) .LS. (8+CYC1) $,     ... PROP. CODE BIT INTO SIGN BIT//
                              IF K LES 0 THEN PUT(PYLET(CYC1),BUFOUT) ... IF ON, INSERT PROPER LETTER//
                              ELSE PUT(40C,BUFOUT) $,       ... IF OFF, INSERT SPACE//
                              INC(BUFOUT,1) $,
                              END $,
          PUT(40C,BUFOUT) $, ... INSERT 2 SPACES BETWEEN REFS.//
          INC(BUFOUT,1) $,
          PUT(40C,BUFOUT) $,
          INC(BUFOUT,1) $,
          END $,
DEFINE INTEGER PROCEDURE GROUP(ENDAD,NOENDS) WHERE INTEGER ENDAD,NOENDS TOBE...MAKE A LIST OF END-CODES FROM ALL STRING//
          BEGIN
          INTEGER GRPAD,CYC1,PART1,PART2,K,CURAD $,
          TOTCOD=NOENDS $,
          TOTWRD=TOTCOD*2 $,
          PART2=(ENDAD+TOTWRD)-(LOC SECT+SIZSEC) $, ... COMPUTE POSSIBLE OVERFLOW OF AFFIX INTO NEXT SECT.//
G1 $      IF PART2 GRT 0 THEN
                              BEGIN
                              PRE=PART2+1 $,
                              TEMP=FREZ(TOTWRD) $,          ... GET STOR. FOR SAVING AFFIXES//
                              PART1=TOTWRD-PART2-1 $,       ... AFFIX WORDS AT END OF THIS SECT.//
                              FOR K=0 STEP 1 UNTIL PART1 DO
                                        W(TEMP+K)=W(ENDAD+K) $,     ... MOVE 1ST PART OF AFFIX LIST//
G2 $                          BEREAD(NAM1,NAM2,SECT TO PRE,OFFND,FOFCNT,RERR) $,  ... READ REST OF AFFIXES//
                              PART1=K-1 $,
                              FOR K=1 STEP 1 UNTIL PART2 DO
                                        W(TEMP+PART1+K)=SECT(K) $,    ... MOVE REST OF AFFIX LIST//
                              ENDAD=TEMP $,
                              END $,
G2.5 $    IF TAGCOM(ENDAD) NEQ 0 THEN GOTO RETURN $,      ... NO NEED TO GROUP INITIALS//
          K=0 $,
          CODES=FREZ(TOTCOD) $,
          GRPAD=ENDAD $,
G3 $      FOR CYC1=1 STEP 1 UNTIL NOENDS DO       ... ENDING-STRINGS-IN-LIST CYCLE//
                              BEGIN
                              L1 $ CURAD=BYTE1(GRPAD+1) $,  ... TAKE LEFT CODE//
                              BYTE1(CODES+K)=CURAD $,       ... INTO LEFT BYTE OF LIST//
               K=K+1 $,
                              GRPAD=GRPAD+2 $,    ... COMPUTE ADDRESS OF NEXT ENDING STRING//
                              END $,
          DECCOM(LOC CODES)=K $,        ... RETURN NO. OF CODES GROUPED//
          GROUP=CODES $,
          END $,
END FINI
```